INT 21: DOS functions

DOS 1.x

AH=00: **Terminate Program** CS->PSP	(=INT 20) restore INT 22,23,24 vectors from old values stored in PSP. does not close open files.
AH=01: **Read Keyboard with Echo** - input may be redirected	AL=InputChar =0 for extended keys (function keys etc) next call reads the scancode <ctrl-break>: INT 23
AH=02: **Display Character** - output may be redirected DL=OutputChar =7 (Beep) <ctrl-break>: INT 23	
AH=03: **Auxiliary Input** wait for input from standard device (COM1) - input may be redirected to other serial port (via MODE)	AL=InputChar <ctrl-break>: INT 23

AH=04: **Auxiliary Output**
to standard device (COM1) - may be redirected
DL=OutputChar
<ctrl-break>: INT 23	

AH=05: **Print Character**
DL=OutputChar - to standard printer (LPT1)
<ctrl-break>: INT 23
	see INT 17

AH=06: **Direct Console I/O** DL= FF:read from standard input (keyboard) - no wait - no echo - no check for <^break> < FF:write DL to standard output (screen)	read - character ready: Zf=0, AL=InputChar (=0: extended key - read again) read - character not ready: Zf=1, AL undefined
AH=07: **Direct Console Input** read from standard input (keyboard) - wait for key - no echo - no check for <^break>	AL=InputChar (=0: extended key - read again)
AH=08: **Read Keyboard without Echo** read from standard input - wait for key - no echo - check for <^break> (INT 23)	AL=InputChar (=0: extended key - read again)
AH=09: **Display String** DS:DX -> String$ each char is sent as with function 02: Display character	
AH=0A: **Buffered Keyboard Input** read string from standard input - wait for input - echo - terminated by <enter> (used by COMMAND.COM at prompt) DS:DX -> buffer: byte[DX] : buffer length (1..255)	byte[DX+1]=bytes read (excl. terminal CR) [DX+2]=first byte terminal byte = 0D <Enter> if buffer is full: further input will be ignored and beeps emitted
AH=0B: **check keyboard status**	AL=00: buffer empty =FF: at least one character in buffer <ctrl-break>: INT 23
AH=0C: **Flush Buffer, Read Keyboard** empty standard input buffer, then: AL=01: INT 21,1: Read Keyboard with Echo =06: INT 21,6: Direct Console I/O =07: INT 21,7: Direct Console Input =08: INT 21,8: Read Keyboard without Echo =0A: INT 21,A: Buffered Keyboard Input	AL=InputChar

AH=0D: **Reset disk drive** ! files with changed size must be closed before the call ! - Flush all file buffers.	

AH=0E: **select disk drive**
selected drive will be new default
DL=driveID: 0=A, 1=B,,,
	AL=total no. of logical drives

AH=0F: **Open File** DS:DX->FCB to be opened	AL=00: file opened - FCB filled =FF: error
AH=10: **Close File** DS:DX->open FCB	AL=00: file closed =FF: error
AH=11: **Search first file match** to filespec (wildcards ok) in FCB. Searches only current directory - for path-search, use INT 21,4E. Before this call: use INT 21,1A: Set DTA. DS:DX->FCB (not open)	AL=00: match found DTA filled with unopened FCB - opening the file destroys info for 'Search next' =FF: not found
AH=12: **Search next file match** Before this call: use previous DS:DX->FCB (not open)	AL=00: match found DTA filled with unopened FCB - opening the file destroys info for 'Search next' =FF: not found
AH=13: **Delete file** All files matching filespec in FCB are deleted. DS:DX->FCB (not open)	AL=00: file(s) deleted =FF: error
AH=14: **Read File Sequential** DS:DX->open FCB with - filename - record size - record counter	AL=00: file record read into memory at DTA FCB updated to next record =01: EOF (no data) =02: DTA too small =03: EOF (part of record read)
AH=15: **Write File Sequential** DS:DX->open FCB with - filename - record size - record counter	AL=00: file record written from DTA to buffer - whole sectors are written to disk FCB updated to next record =01: Disk full =02: DTA too small
AH=16: **Create File** DS:DX->FCB (not open) with filename	AL=00: File created with length 0 =FF: error
AH=17: **Rename File with FCB** DS:DX->FCB with new filespec	AL=00: File(s) renamed =FF: error
AH=19: **Get Default Drive**	AL=0,1,..: A,B,..
AH=1A: **Set DTA-address** (80h in PSP if not set) DS:DX->DTA-address	

AH=1B: **Get FAT info**
for current (default) drive
	AL=sectors pr cluster
	CX=sector size
	DX=number of clusters
	DS:BX->media descriptor byte

AH=1C: **Get FAT info for specified drive**
DL=Drive number
	AL=sectors pr cluster
	CX=sector size
	DX=number of clusters
	DS:BX->media descriptor byte

AH=21: **Read Random Record** DS:DX->open FCB	AL=00: Record read to DTA =01: EOF (no data) =02: DTA too small =03: EOF (part of record read)
AH=22: **Write Random Record** DS:DX->open FCB	AL=00: Record written from DTA =01: Disk full =02: DTA too small

AH=23: **Get File Size** no. of records DS:DX->FCB with - filespec - RecordSize	AL=00: no. of records returned in FCB =FF: error
AH=24: **Set Random Record Number** using CurBlockNo & CurRecNo DS:DX->open FCB	

AH=25: **set interrupt vector**
AL=int no.
DS:DX=new vector	

AH=26: **Create new PSP** with copy of calling .COM programs PSP (CS must point to this) DX=segment addr of new PSP	
AH=27: **Random Block Read** DS:DX->open FCB CX=number of records to be read	AL=00: Records read to DTA =01: EOF (no data) =02: DTA too small =03: EOF (part of record read) CX=number of records read
AH=28: **Random Block Write** DS:DX->open FCB CX=number of records to be written	AL=00: Records written from DTA =01: EOF (no data) =02: DTA too small CX=number of records written
AH=29: **Parse Filename** AL=Parsing control bits: see MS-DOS PR p.254 DS:SI->Command Line ("d:filename.ext'0'") wildcards ok, or several filenames separated by spaces ES:DI->FCB to be filled (not opened)	AL=00: no wildcards used =01: wildcards used =FF: drive not accepted DS:SI->1.char after filename ES:DI->1.byte of filled FCB

AH=2A: **Get Date**	AL=weekday (0=sunday)
	CX=year (1980-2099)
	DH=month (1-12)
	DL=day (1-31)

AH=2B: **Set Date**
CX=year (1980-2099)
DH=month (1-12)
DL=day (1-31)
	AL=00: no error
	AL=FF: error

AH=2C: **Get Time**	CH=hours (0-23)
	CL=minutes (0-59)
	DH=seconds (0-59)
	DL=sec/100 (0-99)

AH=2D: **Set Time**
CH=hours (0-23)
CL=minutes (0-59)
DH=seconds (0-59)
DL=sec/100 (0-99)
	AL=00: no error
	AL=FF: error

AH=2E: **Set Verify On/Off**
AL=0,1: set verify off,on	

DOS 2.x

AH=2F: **Get DTA-address**	ES:BX->DTA

AH=30: **get DOS version number**	AH=minor number
	AL=major number

AH=31: **Keep Program**
AL=return code
DX=size in paragraphs of code to remain in memory	

AH=33: **Get/Set Ctrl-C check flag**
AL=00: Get
AL=01: Set:
DL=00,01: Set off, Set on
	Get:
	DL=00,01: flag is off,on

AH=35: **get interrupt vector**
AL=int no.
	ES:BX=vector

AH=36: **Get Disk Free Space**
DL=drive no. (0=default, 1=A,..)
	AX=no. of sectors pr cluster
	=FFFF: error
	BX=no. of free clusters
	CX=no.of bytes pr sector
	DX=no. of clusters

AH=38: **Set Country Code**
DX=FFFF
IF CountryCode < 254:
AL=CountryCode (45=DK)
ELSE:
AL=FF
BX=CountryCode	

AH=38: **Get Country Info**
DX<>FFFF:
DS:DX->CountryInfo structure
IF CountryCode < 254:
AL=CountryCode (45=DK
0=current country)
ELSE:
AL=FF
BX=CountryCode
	CF=1: error
	AX=error code
	CF=0:
	AL=CountryCode (low 8 bits) (45=DK)
	BX=CountryCode
	DS:DX->filled CountryInfo structure

AH=39: **Create Directory** DS:DX->ASCIIZ DirSpec	CF=1: AX=error code
AH=3A: **Remove Directory** DS:DX->ASCIIZ DirSpec	CF=1: AX=error code
AH=3B: **change current directory** DS:DX->path0	CF=1: AX=error code

AH=3C: **Create File with Handle**
(compare INT 21,5B)
DS:DX->path0
CX=file attrib
	IF file exists: no error
	CF=1: AX=error code
	CF=0: AX=Handle
	file opened

AH=3D: **Open File with Handle**
AL=access code: 0=read
1=write
2=read/write
DS:DX->FileSpec0
	CF=1: AX=error code
	CF=0: AX=Handle
	file opened

AH=3E: **close handle**
BX=handle
	CF=1: AX=error code

AH=3F: **read from file or device**
BX=handle (0=standard input: keybd)
CX=byte count
DS:DX->buffer
	CF=1: AX=error code
	CF=0: AX=bytes read

AH=40: **write to file or device**
BX=handle (1=standard output: scrn)
CX=byte count
DS:DX->buffer
	CF=1: AX=error code
	CF=0: AX=bytes written

AH=41: **Delete File** DS:DX->FileSpec0	CF=1: AX=error code

AH=42: **Move File Pointer**
AL=move method
=00: absolute from file start (+)
=01: relative from current position (+-)
=02: absolute from file end (+-)
BX=handle
CX,DX=move distance (bytes)
	CF=1: AX=error code
	CF=0: DX,AX=new pointer

AX=4300: **Get File Attributes**
DS:DX-> full ASCIIZ filename
	CF=1: AX=error code
	CF=0: CX=attributes

AX=4301: S**et File Attributes**
CX=attributes
DS:DX-> full ASCIIZ filename	

AH=44: **I/O Device Control**
AL=00: Get device info
  =01: Set device info
  =02: Read from character device
  =03: Write to character device
  =04: Read from block device
  etc...
	CF=1: AX=error code
	CF=0: DX=device info (for AL=00)

AH=45: **Duplicate File Handle**
BX=handle
	CF=1: AX=error code
	CF=0: AX=new handle

AH=46: **Force Duplicate File Handle**
BX=handle
CX=new handle
	CF=1: AX=error code

AH=47: **Get Current Directory**
DL=drive no.
DS:SI->64 byte area
	CF=1: AX=error code
	CF=0: DS:SI->filled 64 byte area

AH=48: **Allocate Memory**
BX=requested paragraphs
	CF=1: AX=error code
	CF=0: AX:00->allocated memory
	      BX=allocated paragraphs

AH=49: **Free Allocated Memory** ES=segment of block to free	CF=1: AX=error code

AH=4A: **Set Memory Block Size**
ES=segment of block to set (change)
BX=requested paragraphs
	CF=1: AX=error code
	CF=0: BX=allocated paragraphs

AH=4B: **Load (and Execute) Program**
AL=00: Load and execute
  =03: Load overlay
DS:DX->ASCIIZ filespec
ES:BX->parameter block
	CF=1: AX=error code

AH=4C: **terminate program**
AL='errornr' (used by ERRORLEVEL)	

AH=4D: **Get Return Value**
       from INT 21,31 or INT 21,4C
       - only once!
	AH=return value
	AL=00: normal termination
	  =01: ctrl-break termination
	  =02: critical error
	  =03: termination by INT 21,31

AH=4E: **Find First File**
CX=attrib
DS:DX->ASCIIZ filespec
	CF=1: AX=error code
	CF=0: DTA filled with Fileinfo struc.

AH=4F: **Find Next File**
       DTA contains Fileinfo
	CF=1: AX=error code
	CF=0: DTA Fileinfo updated

AH=52: **get list of lists** (undoc) -PCM.10.24.466:CDS structure -GOOKIN: Enhanced...p325	ES:BX point to system info +16h point to first CDS +21h number of CDS structures set by LASTDRIVE - default: 5 (E)

AH=54: **Get Verify State**	AL=00: Verify is off
	  =01: Verify is on

AH=56: **Rename File** - may be used to 'move' a file DS:DX->ASCIIZ filespec ES:DI->ASCIIZ new filespec	CF=1: AX=error code

AX=5700: **Get File Date and Time**
BX=Handle
	CF=1: AX=error code
	CF=0: CX=time (bits:)
	       15-11: hour (0-23)
	       10- 5: minute (0-59)
	        4- 0: 2sec (0-29)
	      DX=date (bits:)
	       15- 9: year (since 1980)
	        8- 5: month (1-12)
	        4- 0: day (1-31)

AX=5701: **Set File Date and Time**
BX=Handle
CX=time, DX=date (as above)
	CF=1: AX=error code

DOS 3.x

AH=59: **Get Extended Error Info**	AX=extended error code
	>PC'ere UL s.185
	BH=error type
	BL=suggested action
	CH=location of error

AH=5A: **Create 'Temporary' File**
CX=attrib
DS:DX->ASCIIZ dirspec (with terminal'\')
	CF=1: AX=error code
	CF=0: DS:DX->ASCIIZ filespec with filename supplied by DOS

AH=5B: **Create New File**
(compare INT 21,3C)
CX=attrib
DS:DX->ASCIIZ filespec
	IF file exists: error (CF=1)
	CF=1: AX=error code
	CF=0: AX=handle
	file opened

AH=5C: **Lock/Unlock File** AL=00: Lock =01: Unlock BX : Handle CX,DX: Offset (4 byte) SI,DI: Length (4 byte)	CF=1: AX=error code

AX=5E00: **Get Machine Name**
DS:DX->15 byte area
	CF=1: AX=error code
	CF=0:
	CH=0: Name not defined
	CH<>0:
	CL='Name of NETBIOS' (?)
	DS:DX->15 byte ASCIIZ Machine name

AX=5E02: **Set Printer Setup**
BX=index
CX=length of initialize string
DS:DI->initialize string to be sent ahead of all files sent to printer
	CF=1: AX=error code

AX=5E03: **Get Printer Setup**
BX=index
DS:SI->buffer
	CF=1: AX=error code
	CF=0: CX=length of initialize string
	DS:DI->initialize string

AX=5F02: **Get Assign-List Entry** BX=index (no. of current redirection) =0: initial index, to be incremented until AX returns error code 18d DS:SI->16 byte buffer ES:DI->128 byte buffer	CF=1: AX=error code =18d: end of redirection list CF=0: BH=status flag: bit 0=0: device unavailable =1: device available BL=device type =3: printer =4: file CX=parameter value DS:SI->ASCIIZ local device name (PRN, LPT1, ...) ES:DI->ASCIIZ network name

AX=5F03: **Make Network Connection**
BL=3: to printer
=4: to disk drive
CX=parameter (=00)
DS:SI->ASCIIZ device name (PRN, LPT1, ...)
ES:DI->ASCIIZ network path
	CF=1: AX=error code

AX=5F04: **Delete Network Connection** DS:SI->ASCIIZ device name and path	CF=1: AX=error code
AH=62: **Get PSP address**	BX=segment address

DOS 3.3:

AH=65: **Get Extended National Info**
AL=info type
BX=code page
CX=no. of data to be returned
DX=national ID
ES:DI->buffer
	ES:DI->tabel with extended national info

AX=6601: **Get Code Page**	CF=1: AX=error code
	CF=0: BX=active code page
	DX=system code page

AX=6602: **Change Code Page** like DOS command :>CHCP needs installed NLSFUNC	
AH=67: **Set Maximal Handle Count** BX=maximal open files handle count <20: =20	CF=1: AX=error code

AH=68: **Commit File**
Empty file buffer
BX=handle	