MS-DOS OS Function Reference
DOS I/O Function Calls

The following includes most MS-DOS INT 21H functions related to simple file anc console I/O, as well as a few functions related to system date and time.

The listings show what parameters must be supplied before calling the function. In all cases AH must be loaded with the function number before calling INT 21H. Most parameters are passed in registers. In a few cases, specific areas of memory are described as parameters as well.

FUNCTION 01H: Read Keyboard and Echo

Waits for char from standard input, echoes it to standard output and returns char in AL.

Parameters: NONE

Returns:

AL 8-bit char code

Notes:

1. Characters echoed to display; Ctrl-C enabled; waits for input
2. Returns either 8-bit ASCII or one of two bytes of IBM extended ASCII codes

FUNCTION 02H: Display Character

Sends a character to standard output

Parameters:

DL Char to display

Returns: NONE

Notes:

1. Cursor position is updated; if char is backspace 08H cursor moves left but character not erased

FUNCTION 06H: Direct Console I/O

Reads a char from standard input or sends a char to standard output

Parameters:

DL 8-bit char or 0FFH

Returns:

AL 8-bit char or 00H

Flags: ZF set

Notes:

1. If DL = FFH, console input is performed else DL is output to console
2. If ZF clear, AL contains char from console else 00
3. Does not wait for character from standard input

FUNCTION 07H: Direct Console Input without Echo

Waits for char from standard input; does not echo char and ignores Ctrl-C

Parameters: NONE

Returns:

AL 8-bit char from CON

Notes:

1. Does not echo char or check for Ctrl-C

FUNCTION 09H: Display String

Displays a string of characters to standard output. String must end with '$'. All characters up to but not including the '$' are displayed.

Parameters:

DS:DX Segment:offset of pointer to $-terminated string

Returns: NONE

Notes:

1. Obsolete, function 40H preferred
2. String must be terminated with ASCII char 36 ('$')
   This function cannot display a $!

FUNCTION 2AH: Get System Date

Returns the system date

Parameters: NONE

Returns:

AL	Day of Week (Sunday = 0)	CX	Year (1980-2099)
DH	Month (1-12)	DL	Day (1-31)

FUNCTION 2BH: Set System Date

Sets the system date

Parameters:

CX	Year (1980-2099)		
DH	Month (1-12)	DL	Day (1-31)

Returns:

AL 00H = system date updated; FFH = invalid date

Notes: Year must be 16-bit value in range 1980-2099

FUNCTION 2CH: Get System Time

Returns the system time

Parameters: NONE

Returns:

CH	Hour (0-23)	CL	Minutes (0-59)
DH	Seconds (0-59)	DL	Hundredths (0-99)

FUNCTION 2DH: Set System Time

Sets the system time

Parameters:

CH	Hour (0-23)	CL	Minutes (0-59)
DH	Seconds (0-59)	DL	Hundredths (0-99)

Returns: AL 00H = system time updated; FFH = invalid time

FUNCTION 39H: Create Subdirectory (MKDIR)

Creates a subdirectory using specified path

Parameters:

DS:DX Segment:Offset of ASCIIZ string giving pathname

Returns:

AX Error Code if Carry Set

Flags: CF Set if Error

Notes:

1. Possible errors are 03H (Path not Found) and 05H (Access Denied) See also function 59H Extended Error Information
2. MS-DOS puts . and .. entries in all directories.

FUNCTION 3AH: Remove Subdirectory (RMDIR)

Removes (deletes) specified subdirectory

Parameters:

DS:DX Segment:Offset of ASCIIZ string giving pathname

Returns:

AX Error Code if Carry Set

Flags: CF Set if Error

Notes:

1. Possible errors are 03H (Path not Found), 05H (Access Denied) and 10H Attempt to Remove Current Dir. See also function 59H Extended Error Information

FUNCTION 3BH: Change Current Directory (CHDIR)

Changes current directory of default drive

Parameters:

DS:DX Segment:Offset of ASCIIZ string giving pathname

Returns:

AX Error Code if Carry Set

Flags: CF Set if Error

Notes:

1. Possible error is 03H (Path not Found). See also function 59H Extended Error Information
2. Path string is limited to 64 characters, including separators

FUNCTION 3CH: Create File with Handle

Creates a file with specified attributes and returns 16-bit handle. If file exists it is opened and truncated.

Parameters:

CX		Attribute of created file (00H normal 01H read-only 02H hidden 04H system)
DS:DX		Segment:Offset of ASCIIZ string giving pathname

Returns:

AX		Handle or Error Code if Carry Set
Flags:		CF Set if Error

Notes:

1. Possible errors are 03H (Path not Found), 05H (Access Denied) and 04H Too Many Open Files. See also function 59H Extended Error Information
2. Existing files are truncated to 0 length. See also functions 3DH (Open File) 5AH (Create Temp File) and 5BH (Create New File).
3. File pointer is at position 0 following this call.
4. Other attribute bits other than shown above are ignored.

FUNCTION 3DH: Open File with Handle

Opens a file and returns 16-bit handle for subsequent access. File must exist.

Parameters:

AL	Bits 0-2:	Access Code (000=read only, 001=write only, 010=read/write)
	Bit 3:	Reserved (0)
	Bits 4-6:	Sharing Mode
	Bit 7:	Inheritance Bit
DS:DX		Segment:Offset of ASCIIZ string giving pathname

Returns:

AX		Handle or Error Code if Carry Set
Flags:		CF Set if Error

Notes:

1. Possible errors are 02H (file not Found) 03H (Path not Found), 05H (Access Denied), 04H (Too Many Open Files) and 0CH (Invalid Access Code). See also function 59H Extended Error Information.
2. See references for more information on Sharing Mode and Inheritance Bit. For most file access leave these bits zero. Sharing Mode applies primarily to networks.

FUNCTION 3EH: Close File with Handle

Closes file referenced by handle

Parameters:

BX		Handle Number

Returns:

AX		Error Code if Carry Set
Flags:		CF Set if Error

Notes:

1. Possible error is 06H (Invalid Handle). See also function 59H Get Extended Error Information.
2. Buffers are flushed, FAT and directory information is updated.

FUNCTION 3FH: Read File or Device Using Handle

Reads from file or device specified by handle

Parameters:

BX		Handle Number
CX		Number of Bytes to Read
DS:DX		Segment:Offset of Data Buffer

Returns:

AX		Number of Bytes Read or Error Code if Carry Set
Flags:		CF Set if Error
Buffer:		Data Read

Notes:

1. Possible errors are 05H (Access Denied) and 06H (Invalid Handle). See also function 59H Get Extended Error Information.
2. Data is read from file beginning at current pointer location. After

successful read file pointer is updated to point to the byte following the last byte read.

3. Function 3FH can be used with all handles including the keyboard. When reading from standard input, the function terminates when a carriage return is read. Therefore AX may not match CX.
4. If AX = 00H, then an attempt was made to read past the end of file. If AX < CX then a partial "record" was read.

FUNCTION 40H: Write File or Device Using Handle

Writes to file or device specified by handle

Parameters:

	BX	Handle Number
	CX	Number of Bytes to Write
	DS:DX	Segment:Offset of Data Buffer

Returns:

	AX	Number of Bytes Written or Error Code if Carry Set
	Flags:	CF Set if Error
	Buffer:	Unchanged

Notes:

1. Possible errors are 05H (Access Denied) and 06H (Invalid Handle). See also function 59H Get Extended Error Information.
2. Data is written beginning at current file pointer location. After successful read file pointer is updated to point after last byte written and number of bytes written is in AX.
3. If the handle refers to a disk file and the number of bytes in returned in AX is less than the number of bytes in CX then the disk is full. CARRY FLAG IS NOT SET IN THIS SITUATION.
4. If CX = 0 then the file is either truncated or extended to the current file pointer. FAT is updated accordingly.

FUNCTION 41H: Delete File

Deletes directory and FAT entries of specified file (NOT the data)

Parameters:

	DS:DX	Segment:Offset of pointer to ASCIIZ pathname string

Returns:

	AX	Number of Bytes Written or Error Code if Carry Set
	Flags:	CF Set if Error

Notes:

1. Possible errors are 02H (File not Found) 05H (Access Denied) and 03H (Path not Found). See also function 59H Get Extended Error Information.

FUNCTION 42H: Move File Pointer

Sets position of file pointer for subsequent reads or writes.

Parameters:

	AL	Method Code (00H = Byte offset from beginning of file; 01H = Byte offset from current pointer; 02H = Byte offset from end of file)
	BX	Handle Number
	CX:DX	32-bit offset value to move pointer

Returns:

	DX:AX	Current position of file pointer relative to BOF
	Flags:	CF Set if Error

Notes:

1. Possible errors are 01H (AL invalid) 06H (Invalid Handle). See also function 59H Get Extended Error Information.
2. With Method 00H CX:DX is an unsigned doubleword. With methods 01H and 02H CX:DX is a signed doubleword and can be negative.
3. Specifying Method 00H and offset 0 positions pointer at beginning of file. Specifying Method 02H and offset 0 positions pointer at EOF.
4. Specifying Method 02H and offset 0 returns file size in DX:AX.
5. If file pointer is moved BEFORE beginning of file no error is returned but subsequent file access will produce unexpected errors.

FUNCTION 56H: Rename File

Used to rename and/or "move" a file

Parameters:

DS:DX	Segment:Offset of pointer to old ASCIIZ pathname
ES:DI	Segment:Offset of pointer to new ASCIIZ pathname

Returns:

AL	Error Code if CF Set
Flags:	Carry set if error

Notes:

1. Unlike the DOS RENAME command, this function can change the directory path as well as the name of a file; thus effectively "moving" the file. However, file cannot be moved to new drive.
2. Wildcard characters * and ? may not be used in the pathnames.
3. Cannot be used to rename open files.
4. Possible errors returned are 02H (File Not Found) 03H (Path Not Found) 05H (Access Denied) and 11H (Not the Same Device)

Predefined Handles

Every process gets 5 predefined handles. These are as follows:

Handle	Device Assignment	Default Device
0	Standard Input	Keyboard (CON)
1	Standard Output	Display (CON)
2	Standard Error	Display
3	Standard Auxiliary	AUX
4	Standard Printer	LPT1

Note that each process is allowed up to 20 handles (open files total). This can be changed if needed. See DOS references.

DOS Standard Error Codes Returned from Function Calls

Functions that use CF to return success or failure also return an error code in the AX register. These are the standard error codes returned if CF is set following a call a file handle function.

01H	Invalid Function Number
02H	File Not Found
03H	Path Not Found
04H	Too Many Open Files (No Handles Left)
05H	Access Denied
06H	Invalid Handle
07H	Memory Control Blocks Destroyed
08H	Insufficient Memory
09H	Invalid Memory Block Address
0AH	Invalid Environment
0BH	Invalid Format
0CH	Invalid Access Code
0DH	Invalid Data
0EH	Invalid Drive was Specified
10H	Attempted to Remove Current Directory
11H	Not Same Device
12H	No More Files