CHAPTER 8:
SYSTEM CALLS

Introduction

MOS system calls are primarily supported through the INT 21H instruction, or through a call to MOS's Extended Services function handler. An appropriate function code is supplied in the AH register with the other general purpose registers being used to pass parameters related to the function and to return information back to the application. The carry flag may be set on return to indicate an error, in which case the AX register will contain an error code.

The information for each function presented in this chapter is in the following format:

Interrupt:	The software interrupt to use to make the system call
Function Number:	Number to put in the AH register to identify the desired function
Function Name:	Descriptive name of function
Purpose:	Explanation of function use
Data Passed:	Describes state of registers when making function calls
Data Returned:	Describes state of registers after successful function calls

Error Codes: Codes returned in the AX register in the case of an error. Most functions will set the carry flag if an error occurs. (See function 59H for a list of Standard Error Codes)

Extended ErrorCodes: If a function returns an error and more detailed information is desired, interrupt 21H function 59H may be called. (See function 59H for a list of Extended Error Codes)

NOTE: The extended error codes that will be returned for each standard error code are listed in table format for each function in this chapter.

In addition, some errors will cause the operating system to call the critical error interrupt (24H). For those functions which may cause critical errors, extended error codes (not associated with any specific standard error code) are listed.

These extended error codes will be returned to function 59H (Get Extended Error Code) inside a user installed interrupt 24 handler, or after the user tells the standard MOS interrupt 24 handler to abort the currently running program.

NOTE: Functions 0FH - 17H, 21H - 24H, and 27H - 29H are FCB related file calls. They are supported for compatibility only and are not the recommended method for any file I/O since the handle calls are superior. Therefore, they are not documented in this manual.

Functions 5E00H - 5F04H currently have limited support under MOS. The NETBIOS driver must be loaded to provide this support. A device name is returned for function 5E00 with an "end of list" status being returned for the other functions.

Calls Provided for Compatibility Only

Interrupt 25H

Interrupt 25H is used for sector-level disk reads. It is provided only for compatibility with applications written for previous operating systems. It is not further documented in this reference, and is not available for 80386 native mode applications.

Interrupt 26H

Interrupt 26H is used for sector-level disk writes. It is provided only for compatibility with applications written for previous operating systems. It is not further documented in this reference, and is not available for 80386 native mode applications.

Interrupt 27H

Interrupt 27H is used for terminating a program while leaving it resident in memory. Interrupt 21H function 31H is the preferred method for doing this. Interrupt 27H is provided only for compatibility with applications written for previous operating systems. It is not further documented in this reference, and is not available for 80386 native mode applications.

Interrupt 21H

System calls accessed with the INT 21H instruction are detailed in the following pages. It is recommended that the calling method described in Chapter 13 be used whenever possible.

Making calls to an interrupt vector using the pushf/far call coding approach is more efficient than issuing a software interrupt (e.g. an INT 21h instruction). Note, however, that this calling method cannot be used when coding for the 32 bit protected mode of the 80386.

Except as noted, 8086 applications and 80386 native mode applications are interfaced using the same registers and parameters.

Interrupt 21H

Function Number: **00H**

Function Name: Terminate Program

Purpose: To end program execution

Data Passed:

AH = 00H
CS = program's PSP

Data Returned: None

Error Codes: None

Extended Errors: None

Comments: Function 4CH is the preferred method of termination, and the only method for native-mode applications.

Interrupt 21H

Function Number: **01H**

Function Name: Keyboard Read and Echo

Purpose: Reads a character from the standard input device

Data Passed:

AH = 01H

Data Returned:

AL = Character from standard input

Error Codes: None

Extended Errors: None

Comments: This call checks for Ctrl-Break. If encountered, INT 23H is called. If the call returns a 00H in AL, issue another 01H call to get the extended ASCII character.

Interrupt 21H

Function Number: **02H**

Function Name: Display Character

Purpose: Write a character to the standard output device

Data Passed:

AH = 02H
DL = Character to display

Data Returned: None

Error Codes: None

Extended Errors: None

Comments: This function checks for a Ctrl-Break after displaying the character. If found, issues an INT 23H.

Interrupt 21H

Function Number: **03H**

Function Name: Auxiliary Device Read

Purpose: Reads a character from AUX

Data Passed:

AH = 03H

Data Returned:

AL = character from AUX device

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **04H**

Function Name: Auxiliary Device Write

Purpose: Writes a character to AUX

Data Passed:

AH = 04H
DL = Character to write

Data Returned: None

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **05H**

Function Name: Printer Write

Purpose: Write a character to printer

Data Passed:

AH = 05H
DL = Character to write

Data Returned: None

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **06H**

Function Name: Direct Console I/O

Purpose: Reads a character from standard input if one is ready or writes a character to standard output. If the DL register contains 0FFH, an INPUT is requested, otherwise the data in DL is OUTPUT.

Data Passed:

AH = 06H
DL = 0FFH (For console input)

or

AL = char (Character to be output).

Data Returned: ZF is returned set if input was requested and no character was available. No data is returned on output.

Error codes: None

Extended Errors: None

Comments: If the read call returns 00H in AL, issue another red call to get the extended ASCII character. This function does not check for Ctrl-C, Ctrl-Break or Ctrl-P. The characters read are not echoed to standard output.

Interrupt 21H

Function Number: **07H**

Function Name: Direct Console Input - No Echo

Purpose: Reads a character from standard input but does not echo to standard output. Waits if no character is ready.

Data Passed:

AH = 07H

Data Returned:

AL = character from standard input

Error Codes: None

Extended Errors: None

Comments: This function does not check for any special characters.

Interrupt 21H

Function Number: **08H**

Function Name: Read Keyboard - No Echo

Purpose: Reads a character from standard input. Waits if none is ready.

Data Passed:

AH = 08H

Data Returned:

AL = character from standard input

Error Codes: None

Extended Errors: None

Comments: The same as function 7 except this function checks for special characters, such as Ctrl-Break, Ctrl-P. Two calls are required for extended ASCII codes as per functions 1,6.

Interrupt 21H

Function Number: **09H**

Function Name: String Write

Purpose: Writes a string of characters to standard output.

Data Passed:

AH = 09H
DS:DX = address of string.

Data Returned: None

Error Codes: None

Extended Errors: None

Comments: The string must terminate with a 24H($).

Interrupt 21H

Function Number: **0AH**

Function Name: Buffered Keyboard Read

Purpose: Reads characters from standard input into user specified buffer.

Data Passed:

AH = 0AH
DS:DX = buffer address

Data Returned: None

Error Codes: None

Extended Errors: None

Comments: The buffer has two significant bytes at its beginning. The first byte is the number of characters the buffer can contain. The second byte is set by the call. It's a count of the actual number of characters read.

Input is terminated by a carriage return, which is never placed in the buffer, nor accounted for in the read count. If the buffer fills and input continues, the additional characters are ignored, and the bell is rung for each character typed. All of the features of MOS's command line editing are available when using this call.

Interrupt 21H

Function Number: **0BH**

Function Name: Check Keyboard Status

Purpose: Queries standard input for characters

Data Passed:

AH = 0BH

Data Returned:

AL = 0FFH - standard input has a character ready to read.
AL = 00H - No characters ready.

Error Codes: None

Extended Errors: None

Comments: If a Ctrl-Break is encountered, an INT 23H is issued.

Interrupt 21H

Function Number: **0CH**

Function Name: Flush Buffer - Read Keyboard

Purpose: To clear the keyboard buffer and call another standard input read function.

Data Passed:

AH = 0CH
AL = Function to call.

Data Returned: None

Error Codes: None

Extended Errors: None

Comments: Valid function numbers to call are:01H, 06H, 07H, 08H, 0AH.

Interrupt 21H

Function Number: **0DH**

Function Name: Disk Buffer Flush

Purpose: Flush all disk buffers to disk

Data Passed:

AH = 0DH

Data Returned: None

Error Codes: None

Extended Errors: None

Comments: This call will not update directory entries, just the data contained in any open files. To update directory entries you must close the files they correspond to.

Interrupt 21H

Function Number: **0EH**

Function Name: Set Default Disk

Purpose: Sets the default disk

Data Passed:

AH = 0EH
DL = disk number (0 = A, 1 = B, etc.)

Data Returned:

AL = Total number of logical disks.

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **19H**

Function Name: Get Current Drive

Purpose: Determines the current disk drive letter

Data Passed:

AH = 19H

Data Returned:

AL = 0 if drive A, 1 if B, etc.

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **1AH**

Function Name: Set Disk Transfer Area (DTA) Address

Purpose: Sets the address of the Disk Transfer Area (DTA), which is required for functions 4EH and 4FH.

Data Passed:

(8086 Mode) AH = 1AH
(8086 Mode) DS:DX = Desired DTA address

Data Returned: Not Applicable

Error Codes: None

Extended Errors: None

Comments: The DTA address must be such that the DTA will not cross a segment boundary. This function is not needed nor supported for Native Mode applications.

Interrupt 21H

Function Number: **1CH**

Function Name: Get Disk Allocation Information

Purpose: Indicates how the specified disk was formatted

Data Passed:

AH = 1CH
DL = 0 for drive A, 1 for B, etc.

Data Returned:

(Native Mode) DS:EBX = Address of media descriptor byte
(8086 Mode) DS:BX = Address of media descriptor byte
AL = 0FFH if error (invalid drive), otherwise number of sectors per cluster
CX = Number of bytes per sector
DX = Number of clusters allocated

Error Codes: None

Extended Errors: None

Comments: See also function 36H.

Interrupt 21H

Function Number: **25H**

Function Name: Set Interrupt Vector

Purpose: Sets an interrupt vector

Data Passed:

AH = 25H
DS:DX = Address of interrupt service routine
AL = Interrupt number

Data Returned: None

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **2AH**

Function Name: Get System Date

Purpose: Returns the current date

Data Passed:

AH = 2AH

Data Returned:

AL = Day of week (0=Sunday, 1=Monday, etc.)
CX = Year (1980-2099)
DH = Month (1-12)
DL = Day of month (1-31)

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **2BH**

Function Name: Set System Date

Purpose: Set the current date

Data Passed:

AH = 2BH
CX = Year (e.g. 1990)
DH = Month (1-12)
DL = Day of month (1-31)

Data Returned:

AL = Status. (0FFH = invalid date)

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **2CH**

Function Name: Get Time of Day

Purpose: Returns the current time of day

Data Passed:

AH = 2CH

Data Returned:

CH = Hours (0-23)
CL = Minutes (0-59)
DH = Seconds (0-59)
DL = Hundredths of seconds (0-99)

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **2DH**

Function Name: Set Time of Day

Purpose: Sets the current time of day

Data Passed:

AH = 2DH
CH = Hours (0-23)
CL = Minutes (0-59)
DH = Seconds (0-59)
DL = Hundredths of seconds (0-99)

Data Returned:

AL = OFF if invalid time, otherwise 0

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **2EH**

Function Name: Verify On/Off

Purpose: Tells the system if disk writes should be verified

Data Passed:

AH = 2EH
AL = 1 to verify, 0 to not verify

Data Returned: Not Applicable

Error Codes: None

Extended Errors: None

Comments: See also function 54H.

Interrupt 21H

Function Number: **2FH**

Function Name: Get Disk Transfer Address

Purpose: Retrieve the current Disk Transfer Address

Data Passed:

AH = 2FH

Data Returned:

ES:BX = Current DTA

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **30H**

Function Name: Get Version Number

Purpose: Compatibility

Data Passed:

AH = 30H
AX = BX = CX = DX for the MOS version number, otherwise the DOS equivalent version number is returned.

Data Returned:

AL = The major version number
AH = The minor version number
BX = 0
CX = 0

Error Codes: None

Extended Errors: None

Comments: This function is to provide compatibility with many applications that existed prior to MOS. These applications check the version/revision of the operating system to determine if needed features are supported.

To get the PC-MOS version (in AX), set AX=BX=CX=DX before calling the function.

Interrupt 21H

Function Number: **31H**

Function Name: TSR - Terminate and Stay Resident

Purpose: Terminates a process and leaves it in memory.

Data Passed:

AH = 31H
AL = Process return code
DX = Number of paragraphs to leave resident

Data Returned: None

Error Codes: None

Extended Errors: None

Comments: This call is the recommend method for TSR programs to go resident. The return code is passed to the parent process if the parent issues a function 4DH (get subprocess return code).

Open files are NOT closed by this function. The initialization routine must close all files in use before issuing this call.

Interrupt 21H

Function Number: 33H

Function Name: Get/Set Break Switch

Purpose: Get or Set the "break" status flag

Data Passed:

AH = 33H
AL = 0 - to GET the break status
1 - to SET the break status
DL = 0 - BREAK OFF
1 - BREAK ON

Data Returned:

DL = Current break status for function 0

Error Codes:

AL = 0FFH if the wrong function code is input in the AL register

Extended Errors: None

Comments: The "break condition" here refers to whether or not the operating system takes action if keyboard input is not being requested, but the workstation operator keys a Ctrl-Break.

Interrupt 21H

Function Number: **34H**

Function Name: Get "INMOS" flag pointer

Purpose: Returns a pointer to the "inmos" flag. Also provides applications with the segment address of the SCB.

Data Passed:

AH = 34H

Data Returned:

ES:BX = pointer to "INMOS" flag

Error Codes: None

Extended Errors: None

Comment: The "INMOS" flag is a byte which indicates if MOS is using its stack or if it has a task's stack. If the flag is 0, MOS has a task's stack and it is safe to call MOS. Otherwise, if non-zero, MOS is using its own stack and should not be called.

Having the segment address of the SCB provides the application the ability to retrieve MOS's Extended Services entry point. See the Extended Services section of this chapter for details.

Interrupt 21H

Function Number: 35H

Function Name: Get Interrupt Vector

Purpose: Retrieves current vector pointer for specified vector number

Data Passed:

AH = 35H
AL = Vector number to retrieve

Data Returned:

ES:BX = current interrupt vector pointer.

Error Codes: None

Extended Errors: None

Interrupt 21H

Function Number: **36H**

Function Name: Get Disk Space Information

Purpose: Gets the amount of available space on the specified drive

Data Passed:

AH = 36H
DL = 0 for default drive, 1 for A, ...

Data Returned:

AX = Number of sectors per cluster (FFFFH if the specified drive does not exist)
BX = Number of available (unused) clusters
CX = Number of bytes per sector
DX = Number of clusters allocated

Error Codes: Critical Errors

Extended Error Codes:

Error Code	Type	Action	Location
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: See also function 1CH.

Interrupt 21H

Function Number: **38H**

Function Name: Get/Set Country Information

Purpose: To allow utilities to determine how to display country-dependent information.

Data Passed:

AH = 38H

For Get call:

AL = 00H to get current country of the number of the country if < 255. 0FFH - use BX as 16-bit country code if country > or = 255
BX = country code if AL > or = 255
(Native Mode) DS:EDX = Points to caller's buffer to receive information in format shown in table.
(8086 Mode) DS:DX = Points to caller's buffer to receive information in format shown in table.

For Set call:

AL = Country code if code < 255 else 0FFH if code > or = 255
BX = country code if AL = 255
DX = 0FFFFH

Data Returned:

For Get call:

AX = error code if carry
BX = country code
(Native Mode) DS:EDX = Points to buffer of information shown in table.
(8086 Mode) DS:DX = Points to buffer of information shown in table.

For Set call:

AX = error code if carry

Error Codes:

2 - Invalid filename (bad country code)

Extended Error Codes:

Error Code	Type	Action	Location
2	8	3	2

(See function 59H for extended error code meanings)

Comments: This function is used by MOS utilities to determine how information should be displayed. The country information buffer and associated fields are defined below. NOTE: Keyboard support for foreign languages is not necessarily supplied with MOS. See your distributor.

Size	Contents
Word	Binary date. see below
5 bytes	A Null terminated currency symbol string
2 bytes	A Null terminated thousands separator string
2 bytes	A Null terminated decimal separator string
2 bytes	A Null terminated date separator string
2 bytes	A Null terminated time separator string
byte	Currency format indicator. see below
byte	number of significant digits in currency
byte	clock-type indicator. see below
2 words	Case set map for codes > 254. see below
2 bytes	Null terminated data list separator
5 words	Reserved

The date has these forms:

Code	Format
0 = USA	Jan 1 1987
1 = Europe	1 Jan 1987
2 = Japan	1987 Jan 1

The currency indicator has these values:

0 - Currency symbol precedes the amount. No separating space. For example, $10.00.

1 - Currency symbol follows the amount. No separating space. For example, 45c.

2 - Currency symbol precedes the amount. One space separating. For example, $ 10.00.

3 - Currency symbol follows the amount. One space separating. For example 45 c.

4 - Currency symbol replaces the decimal separator.

The clock type indicator:

0 - 12 hour clock.
1 - 24 hour clock.

The Case set call address is used to convert lowercase characters to uppercase, given the current country code. It is a FAR call and AL contains the character to be converted upon entry and the converted character upon return. The codes are ASCII characters.

Interrupt 21H

Function Number: **39H**

Function Name: Make Directory

Purpose: Creates a disk directory

Data Passed:

AH = 39H
(Native Mode) DS:EDX = Address of directory name
(8086 Mode) DS:DX = Address of directory name

Data Returned:

AX = Error code if carry is set

Error Codes:

2 - Invalid filename
3 - Invalid path
5 - Access denied (file or directory can't be read/written to)

Extended Error Codes:

Error Code	Type	Action	Location
2	8	3	2
3	8	3	2
5	3	4	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: The specified directory name must be null-terminated and may include full drive and path specifications, otherwise defaults are assumed. The last directory name in the path is the one created; preceding names must already exist. The string length must not exceed 64 bytes.

Interrupt 21H

Function Number: **3AH**

Function Name: Remove Directory

Purpose: Deletes a directory from disk

Data Passed:

AH = 3AH
(Native Mode) DS:EDX = Address of directory name
(8086 Mode) DS:DX = Address of directory name

Data Returned: AX = Error code if carry is set

Error Codes:

2 - Invalid filename
3 - Invalid path
5 - Access denied
16 - Can't delete current directory

Extended Error Codes:

Error Code	Type	Action	Location
2	8	3	2
3	8	3	2
5	3	4	1
16	3	3	2
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

Comments: See function 39H for name format.

Interrupt 21H

Function Number: **3BH**

Function Name: Set Default Directory

Purpose: Establish the current (default) directory for a particular disk drive

Data Passed:

AH = 3BH
(Native Mode) DS:EDX = Address of directory name
(8086 Mode) DS:DX = Address of directory name

Data Returned: AX = Error code if carry is set

Error Codes:

2 - Invalid filename
3 - Invalid path

Extended Error Codes:

Error Code	Type	Action	Location
2	8	3	2
3	8	3	2
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

Comments: See function 39H for name format. If a drive letter is not specified the current drive is assumed. For each task, the system maintains a current directory for each disk drive letter.

Interrupt 21H

Function Number: **3CH**

Function Name: Create File

Purpose: Creates and opens a file on disk

Data Passed:

AH = 3CH
CX = Attribute bits
(Native Mode) DS:EDX = Address of filename
(8086 Mode) DS:DX = Address of filename

Data Returned:

AX = Error code if carry is set; otherwise a file handle

Error Codes:

3 - Invalid path
4 - Too many files open
5 - Access denied (file or directory can't be read/written to)

Extended Error Codes:

Error Code	Type	Action	Location
3	8	3	2
4	1	4	1
5	3	4	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: See function 39H for the format of the name string. See function 43H for the meanings of the attribute bits. If the file previously existed, then its data is erased. The new file is opened in Compatibility Mode for read/write access (see function 3DH).

Interrupt 21H

Function Number: **3DH**

Function Name: Open File

Purpose: Prepare an existing file for reading or writing

Data Passed:

AH = 3DH
AL = Inheritance, Sharing Mode, Access Bits
(Native Mode) DS:EDX = Address of filename
(8086 Mode) DS:DX = Address of filename

Data Returned:

AX = Error code if carry is set; otherwise a file handle

Error Codes:

3 - Invalid path
4 - Too many files open
5 - Access denied (file or directory can't be read/written to)
32 - Attempt to open file locked by someone else

Extended Error Codes:

Error Code	Type	Action	Location
3	8	3	2
4	1	4	1
5	3	4	1
32	2	2	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: See function 39H for the format of the name string.

The AL value passed has the binary format ISSSXAAA, where:

I = 0 if the file is inherited by child processes, 1 if not;

SSS = Sharing Mode;
000 for Compatibility sharing mode (see below),
001 to disallow sharing,
010 to allow read sharing,
011 to allow write sharing,
100 to allow read/write sharing;

X = Unused bit;

AAA = 000 for Read Access,
001 for Write Access,
010 for Read/Write Access

"Compatibility" sharing mode is provided for compatibility with applications written for older operating systems; it should not be used for new applications because it can cause error messages requiring operator intervention.

The following decision table summarizes the effects of the various sharing modes when a file being opened by task B has already been opened by another task A:

Sharing Mode		TASK B				
		000	001	010	011	100
TASK A	000	OK	Deny	Deny	Deny	Deny
	001	CE	Deny	Deny	Deny	Deny
	010	CE	Deny	R/R	W/R	X/R
	011	CE	Deny	R/W	W/W	X/W
	100	CE	Deny	R/X	W/X	OK

Where:

OK = Task B opens successfully

Deny= Task B open fails with "access denied" return code

CE = Critical Error (INT 24H) occurs in task B, which by default requires operator intervention

R/R = Both tasks must open for read-only access, otherwise access will be denied to task B

R/W = Task A must open read-only and task B write-only, otherwise access will be denied to task B

R/X = Task A must open for read-only access, otherwise access will be denied to task B

W/R = Task A must open write-only and task B read-only, otherwise access will be denied to task B

W/W = Both tasks must open for write-only access, otherwise access will be denied to task B

W/X = Task A must open for write-only access, otherwise access will be denied to task B

X/R = Task B must open for read-only access, otherwise access will be denied to task B

X/W = Task B must open for write-only access, otherwise access will be denied to task B

A successful open sets the file's I/O pointer to zero, which corresponds to the first byte in the file.

Interrupt 21H

Function Number: **3EH**

Function Name: Close File

Purpose: Close a file handle

Data Passed:

AH = 3EH
BX = File handle

Data Returned:

AX = Error code if carry is set

Error Codes:

6 - Invalid handle

Extended Error Codes:

Error Code	Type	Action	Location
6	7	4	1
Critical Error	11	7	2
	11	5	2
	11	1	2
	11	1	1
	9	4	1
	7	4	1
	5	1	1

(See function 59H for Extended Error Code meanings)

Comments: Closing a file causes its buffers to be flushed and the directory entry to be updated (for disk files).