

# Chapter 3

## CP/M System Calls

---

System Call Parameters	44
0 System Reset	45
1 Console Input	46
2 Console Output	47
3 Reader Input	48
4 Punch Output	49
5 List Output	50
6 Direct Console I/O	51
7 Get IOBYTE	52
8 Set IOBYTE	53
9 Print String	55
10 Read Console Buffer	56
11 Get Console Status	58
12 Return Version Number	59
13 Reset Disk System	60
14 Select Disk	61
15 Open File	62
16 Close File	64

- 17 Search for First 65
- 18 Search for Next 66
- 19 Delete File 67
- 20 Read Sequential 68
- 21 Write Sequential 69
- 22 Make File 70
- 23 Rename File 71
- 24 Return Login Vector 72
- 25 Return Current Disk 73
- 26 Set DMA Address 74
- 27 Get Addr Alloc 75
- 28 Write Protect Disk 76
- 29 Get Read/Only Vector 77
- 30 Set File Attributes 78
- 31 Get Addr Disk Parms 79
- 32 Set/Get User Code 80
- 33 Read Random 81
- 34 Write Random 83
- 35 Compute File Size 85
- 36 Set Random Record 87
- 37 Reset Drive 89
- 40 Write Random With Zero Fill 90

This chapter numerically lists the 39 CP/M system calls supported by the Premium SoftCard II System. A listing of the system calls is shown in the following table. Guidelines for using CP/M system calls are given in "Using CP/M System Calls" in Chapter 2.

**Table 3.1.**

**CP/M System Calls Available**

Call Number	Name	Call Number	Name
0	System Reset	20	Read Sequential
1	Console Input	21	Write Sequential
2	Console Output	22	Make File
3	Reader Input	23	Rename File
4	Punch Output	24	Return Login Vector
5	List Output	25	Return Current Disk
6	Direct Console I/O	26	Set DMA Address
7	Get IOBYTE	27	Get Addr Alloc
8	Set IOBYTE	28	Write Protect Disk
9	Print String	29	Get Read/Only Vector
10	Read Console Buffer	30	Set File Attributes
11	Get Console Status	31	Get Addr Disk Params
12	Return Version Number	32	Set/Get User Code
13	Reset Disk System	33	Read Random
14	Select Disk	34	Write Random
15	Open File	35	Compute File Size
16	Close File	36	Set Random Record
17	Search for First	37	Reset Drive
18	Search for Next	40	Write Random With Zero Fill
19	Delete File		

## System Call Parameters

In each of the system call descriptions, a table of parameters shows the required parameters, and into which registers they are loaded. The parameters in each table are:

<i>Entry point</i>	The system call number and the register it is loaded into.
<i>Entry value</i>	The data to be sent to the CPU for processing.
<i>Returned value</i>	The data returned by the CPU as a result of the system call.

For example, the following table shows the value returned in register A which contains either an ASCII character or zero, depending on how the call was executed.

Parameter	Register	Contents
Entry point	C	06H
Entry value	E	0FFH (input) or character (output)
Returned value	A	Character or 00H

In addition to the parameter table, a remarks section describes any special conditions or singularities for using the system call.

## 0 System Reset

### Purpose

Performs a warm start.

### Parameters

Parameter	Register	Contents
Entry point	C	00H
Entry value	None	None
Returned value	None	None

### Remarks

System Reset instructs CP/M to perform a warm start. (This is the same as JMP instruction to location 00H.) Specifically, System Reset performs the following actions:

- Reinitializes the disk drive system by selecting drive A: as the active drive

- Reads the CCP module into memory from the disk in drive A:

- Initializes all I/O devices that have an initialization routine

- Clears the contents of the disk file buffer

- Transfers control to the CCP module

## 1 Console Input

### Purpose

Reads an ASCII character from the logical Console device.

### Parameters

Parameter	Register	Contents
Entry point	C	01H
Entry value	None	None
Returned value	A	Character from the Console device

### Remarks

Console Input reads the next character from the physical device assigned to the Console (CON:) device into register A. If a carriage return, linefeed, backspace, or graphic character is read, Console Input “echoes” the character back to the Console device for display. If a tab character (CONTROL-I) is read, the cursor is moved eight spaces to the next tab stop.

Console Input also checks for CONTROL-S (start/stop scroll), and CONTROL-P (start/stop printer echo). If CONTROL-P is present, all subsequent characters are echoed to the logical LST: device. Control is not returned to the calling program until the next character is entered from the Console device.

A subsequent CONTROL-P will disable echoing of characters to the printer.

## 2 Console Output

### Purpose

Sends an ASCII character to the logical Console device.

### Parameters

Parameter	Register	Contents
Entry point	C	02H
Entry value	E	A character
Returned value	None	None

### Remarks

Console Output sends a character to the logical Console device from register E. If a tab character (CONTROL-I) is sent, up to eight blanks are output to move the cursor to the next tab stop. Console Output also checks for CONTROL-S (start/stop scroll), and CONTROL-P (start/stop printer echo).

### 3 Reader Input

#### Purpose

Reads a character from the current logical Read device (RDR:).

#### Parameters

Parameter	Register	Contents
Entry point	C	03H
Entry value	None	None
Returned value	A	A character

#### Remarks

Reader Input reads into register A the next character from the physical device assigned to RDR:. As in system call 1, Console Input, control is not returned to the calling program until a character has been read.



## 4 Punch Output

### Purpose

Sends an ASCII character to the logical Punch device (PUN:).

### Parameters

Parameter	Register	Contents
Entry point	C	04H
Entry value	E	ASCII character
Returned value	None	None

### Remarks

Punch Output sends an ASCII character to the logical Punch device (PUN:) from register E. Control is not returned to the calling program until the character has been sent.

## 5 List Output

### Purpose

Sends an ASCII character to the logical List device (LST:).

### Parameters

Parameter	Register	Contents
Entry point	C	05H
Entry value	E	ASCII character
Returned value	None	None

### Remarks

List Output sends an ASCII character to the logical List device (LST:) from register A. Control is not returned to the calling program until the character has been sent.

## 6 Direct Console I/O

### Purpose

Initiates direct console I/O.

### Parameters

Parameter	Register	Contents
Entry point	C	06H
Entry value	E	FFH (input) or character (output)
Returned value	A	Character or 00H

### Remarks

Direct Console I/O initiated in register E either contains a value of FFH for console input request, or an ASCII character for output. Upon return, if the value in register E was FFH, register A will contain 00H. Otherwise, register A will contain the next input character from the console.

### Note

We do not recommend using Direct Console I/O, since it bypasses all of CP/M's normal control character functions, such as CONTROL-S and CONTROL-P. Programs which perform direct I/O through the BIOS under previous releases of CP/M, however, should be changed to use direct I/O under BDOS so they can be fully supported under future releases of MP/M<sup>™</sup> and CP/M.

## 7 Get IOBYTE

### Purpose

Returns the current value of the IOBYTE.

### Parameters

Parameter	Register	Contents
Entry point	C	07H
Entry value	None	None
Returned value	A	I/O byte value

### Remarks

The IOBYTE determines the logical to physical device assignment. The IOBYTE value can be displayed at the Console device by using system call 2, Console Output.

## 8 Set IOBYTE

### Purpose

Changes the logical to physical device assignment.

### Parameters

Parameter	Register	Contents
Entry point	C	08H
Entry value	E	New I/O byte value
Returned value	None	None

### Remarks

Set IOBYTE permits changing the IOBYTE value within programs running in the TPA. The IOBYTE format is shown in the following table. Table 3.2 also shows the possible values of the IOBYTE.

**Table 3.2.**  
**IOBYTE Values**

<b>Field (Bits)</b>	<b>Decimal Value</b>	<b>Description</b>
Console	xxx0	TTY: assigned (Default)
	xxx1	CRT: assigned
	xxx2	Batch (BAT:) mode
	xxx3	UC1: assigned
Reader	xx0x	TTY: assigned (Default)
	xx1x	CRT: assigned
	xx2x	PTR: assigned
	xx3x	UR2: assigned
Punch	x0xx	TTY: assigned (Default)
	x1xx	PTP: assigned
	x2xx	UP1: assigned
	x3xx	UP2: assigned
List	0xxx	TTY: assigned
	1xxx	CRT: assigned
	2xxx	LPT: assigned (Default)
	3xxx	UL1: assigned

## 9 Print String

### Purpose

Sends a character string to the logical Console device.

### Parameters

Parameter	Register	Contents
Entry point	C	09H
Entry value	DE	String address
Returned value	None	None

### Remarks

Print String sends a character string from the address contained in register pair DE to the logical Console device. Character strings must end with a "\$" character. If the character string contains tab characters, they are expanded in the same manner as in system call 1, Console Input. Print String also checks for CONTROL-S (start/stop scroll) and for CONTROL-P (printer echo).

## 10 Read Console Buffer

### Purpose

Reads the contents of the Console device buffer.

### Parameters

Parameter	Register	Contents
Entry point	C	0AH
Entry value	DE	Buffer address
Returned value	Buffer	Console characters

### Remarks

Read Console Buffer reads the edited input from the Console logical device into the buffer address specified in register pair DE. (The buffer address is determined by the calling program.) Input is terminated when either the buffer overflows (maximum 255 characters), or a terminating character (carriage return or linefeed) is read into the buffer. The Read Console Buffer is in the following format:

Byte	0	1	2	3	4	5	6	7	8	...	n
Field	mx	nc	c1	c2	c3	c4	c5	c6	c7	...	cn

**Figure 3.1. Console Buffer**

mx equals 255 characters (the buffer's maximum capacity).

nc equals the number of characters read (set by FDOS upon return).

c1—cn equals the characters read from the Console device.



If *nc* is less than *mx*, the uninitialized positions follow the last character (*cn*).

Input to the buffer can be edited with the following line-editing commands:

CONTROL-C	Performs a warm start (if entered at the beginning of line)
CONTROL-E	Denotes the end of the line
CONTROL-H	Backspaces one character position
CONTROL-J	Terminates the input line (linefeed)
CONTROL-M	Terminates the input line (carriage return)
CONTROL-R	Retypes the current line after a new line
CONTROL-X	Backspaces to the beginning of the current line

---

### *Note*

Line editing commands which move the cursor to screen column 0 (e.g., CONTROL-X) will only move the cursor to the column position where the screen prompt ended. This allows for a more legible display. (In earlier CP/M versions, the cursor was returned to the column 0.)

---

## 11 Get Console Status

### Purpose

Monitors the logical Console device for input.

### Parameters

Parameter	Register	Contents
Entry point	C	0BH
Entry value	None	None
Returned value	A	Console status value

### Remarks

If CON: sends a character, register A will contain FFH. Otherwise, register A contains 00H.

## 12 Return Version Number

### Purpose

Returns the CP/M version number.

### Parameters

Parameter	Register	Contents
Entry point	C	0CH
Entry value	None	None
Returned value	HL	Version number

### Remarks

Return Version Number provides a means of programming that is not version dependent. When called, Return Version Number returns a two-byte value representing the version number in register pair HL. The value in register H indicates CP/M (H=00H) or MP/M (H=01H). The value in register L indicates the version of CP/M as follows:

L=00H	All releases prior to 2.0
L=20H	CP/M 2.0
L=21H	CP/M 2.1
L=22H	CP/M 2.2

Return Version Number is useful for writing application which provide both sequential and random access functions. (Random access is disabled for CP/M releases prior to 2.0.)

## 13 Reset Disk System

### Purpose

Resets the disk system from within a program.

### Parameters

Parameter	Register	Contents
Entry point	C	0DH
Entry value	None	None
Returned value	None	None

### Remarks

Reset Disk System resets the disk drive system from within a calling program. This is useful for application programs that require a disk change without a warm or cold start.

When called, Reset Disk System assigns all drives with read or write only attributes and makes disk drive A: the active drive. It also sets the default DMA address to BOOT+0080H. (See system calls 28, Write Protect Disk, and 29, Get Read/Only Vector, for more information on read and write only attributes.)

## 14 Select Disk

### Purpose

Changes the active drive.

### Parameters

Parameter	Register	Contents
Entry point	C	0EH
Entry value	E	Selected disk
Returned value	None	None

### Remarks

Select Disk changes the current active disk drive to the drive represented by the value in register E. The value of register E is as follows:

E=00H    Drive A:  
E=01H    Drive B:  
E=02H    Drive C:  
E=03H    Drive D:

When selected, the active drive is placed "on-line," which activates its directory in memory until the next cold start, warm start, or disk system reset operation is performed. If the disk is changed while it is on-line, the drive's status is changed to read/only status (see system call 29, Get Read/Only Vector).

During file operations, an FCB which contains 00 for the drive code will automatically access the active drive. Drive codes one through three ignore the active drive and access the selected drive (A: through D:).

## 15 Open File

### Purpose

Opens an existing file.

### Parameters

Parameter	Register	Contents
Entry point	C	0FH
Entry value	DE	FCB address
Returned value	A	Directory code

### Remarks

Open File searches the disk directory in the current user area for a filename that matches the name in the FCB contained in register pair DE. Wild card characters (?) and (\*) can be used in the fn and type fields of the FCB. If no wild card characters are included, bytes ex and s2 of the FCB are set to zero. See "The File Control Block" in Chapter 1 for a description of the FCB fields.

If the FCBs match, the relevant file directory information from the disk is copied into the d0—dn field of the addressed FCB. This allows access to the file through subsequent read and write system calls.

---

**Note**

Existing files should not be accessed until a successful open operation is completed.

---

When a file has been opened, Open File returns the directory code with the value zero through three in register A. Otherwise, Open File returns 0FF in register A. If there are wild card characters in the addressed FCB, then the first matching directory FCB is selected. If the file is to be accessed sequentially from the first record, the calling program must set the current record (cr) field to zero.

## 16 Close File

### Purpose

Closes an existing open file.

### Parameters

Parameter	Register	Contents
Entry point	C	10H
Entry value	DE	FCB address
Returned value	A	Directory code

### Remarks

Close File closes an existing open disk file in the current user area. If the file was opened using an Open File or Make File system call, Close File records the file's new FCB information in the referenced disk directory. The FCB matching process for the Close File system call is identical to the Open File system call. When a file is closed, the directory code in register A is 0H, 1H, 2H, or 3H. Otherwise, 0FFH is returned if the filename cannot be found in the directory.

If a file has been written to, it must be closed in order to update the FCB of the file. A file need not be closed for read operations.



## 17 Search for First

### Purpose

Searches for the first file match.

### Parameters

Parameter	Register	Contents
Entry point	C	11H
Entry value	DE	FCB address
Returned value	A	Directory code

### Remarks

Search for First searches the disk directory of the current active user area for the first filename matched by the addressed FCB. If found, Search for First returns a value between 0H and 3H in register A. Otherwise, FFH is returned if the file is not found.

When the addressed file is found, Search for First writes the matching directory entry and the relative starting position into the current DMA address. This is not normally required for application programs, but it permits the directory information to be obtained from the DMA buffer by the calling program.

The ? wild card character can be used in FCB fields f1—f8, t1—tn, and ex to match the corresponding field of a directory entry on the active drive. If the dr field contains a question mark (?), however, the active disk select function is disabled and the default drive is searched. Search for First will then return any matched entry, allocated or free, belonging to any user number. This is not normally used by application programs, but permits greater flexibility to search all current directory entries. If the dr field is not a question mark, the s2 byte is automatically set to zero.

## 18 Search for Next

### Purpose

Searches for the next file match.

### Parameters

Parameter	Register	Contents
Entry point	C	12H
Entry value	None	None
Returned value	A	Directory code

### Remarks

Search for Next is similar to the Search for First system call, except that the directory search continues from the last matched entry. If a match is found, Search for Next returns a value between 0H and 3H in register A. 0FFH is returned when no more directory items match.

## 19 Delete File

### Purpose

Deletes a file or files.

### Parameters

Parameter	Register	Contents
Entry point	C	13H
Entry value	DE	FCB address
Returned value	A	Directory code

### Remarks

The FCB in register pair DE may contain wild card characters (? or \*) in the f1—f8 and t1—t3 fields, but not in the dr field (as in the Search for First and Search for Next system calls).

If the file(s) exist and can be deleted, Delete File returns a value between 0H and 3H in register A. If the file(s) cannot be found, a value of 0FFH is returned.

## 20 Read Sequential

### Purpose

Reads a record sequentially.

### Parameters

Parameter	Register	Contents
Entry point	C	14H
Entry value	DE	FCB address
Returned value	A	Directory code

### Remarks

Read Sequential reads the next 128-byte record from the addressed file into memory at the current DMA address. Before Read Sequential can be used, the FCB in register pair DE must be activated through system call 15, Open File, or system call 22, Make File. If the FCB is present, Read Sequential reads the next 128-byte record from the file into memory at the current DMA address. The record's location is read from the FCB cr (current record) field of the extent, and the value of cr is automatically incremented to the next record position. If the cr field overflows, the next logical extent is automatically opened and the cr field is reset to zero in preparation for the next read operation.

Read Sequential returns a value of 00H in register A when the operation has been completed, and a non-zero value if no data exists at the next record position until the end of the file is reached.

## 21 Write Sequential

### Purpose

Writes data to a file sequentially.

### Parameters

Parameter	Register	Contents
Entry point	C	15H
Entry value	DE	FCB address
Returned value	A	Directory code

### Remarks

Write Sequential writes the next 128-byte record to the addressed file at the current DMA address. Write Sequential can be used only if the FCB address in register pair DE has been activated through system call 15, Open File, or system call 22, Make File. If the FCB is present, Write Sequential writes the next 128-byte data record to the open file from the current DMA address. The cr field of the addressed FCB is automatically incremented to the next record position. If the cr field overflows, the next logical extent is automatically opened and the cr field is reset to zero to prepare for the next write operation. Records written into an existing file overlay those which already exist in the file.

When the write operation has been completed, Write Sequential returns a value of 00H in register A, or a non-zero value for an unsuccessful write operation due to a full disk.

## 22 Make File

### Purpose

Creates or “makes” a new file.

### Parameters

Parameter	Register	Contents
Entry point	C	16H
Entry value	DE	FCB address
Returned value	A	Directory code

### Remarks

Make File is similar to the Open File system call, except that the FCB must not contain a filename of an existing file in the active disk directory. When executed, Make File also creates the file and initializes both the FCB disk directory and the FCB in memory.

Make File returns a value between 0H and 3H in register A if the file was created, and 0FFH if no more directory space was available to create the file. Make File also activates the FCB, so a subsequent open operation is not necessary for writing to the file.

## 23 Rename File

### Purpose

Renames an existing file.

### Parameters

Parameter	Register	Contents
Entry point	C	17H
Entry value	DE	FCB address
Returned value	A	Directory code

### Remarks

Rename File changes the filename and extension in the first 16 bytes of the addressed FCB to the filename and extension in the second 16 bytes. The FCB drive code (dr) selects the drive, while the drive code for the new filename in d0 (byte 16) is assumed to be 0H.

Rename File returns a value between 0H and 3H in register A when the file is renamed. If the file cannot be renamed, Rename File returns a value of 0FFH.

## 24 Return Login Vector

### Purpose

Writes the CP/M login vector into register pair HL.

### Parameters

Parameter	Register	Contents
Entry point	C	18H
Entry value	None	None
Returned value	HL	Login vector

### Remarks

CP/M returns a 16-bit login vector value in register pair HL. The least significant bit position in register L denotes the first drive (A:), and the most significant bit position in register H denotes the fourth drive (drive D:). "0" bit indicates that the drive is off-line, while a "1" bit indicates the drive is on-line. Drives can be brought on-line by an explicit disk drive selection, or by an implicit drive selection caused by a file operation which specified a non-zero dr field.

### Note

To maintain compatibility with earlier CP/M releases, registers A and L contain the same values upon return of the call.



## 25 Return Current Disk

### Purpose

Indicates the current active drive.

### Parameters

Parameter	Register	Contents
Entry point	C	19H
Entry value	None	None
Returned value	A	Current disk

### Remarks

Return Current Disk returns a value in register A that corresponds to the current active drive. The possible values in register A are as follows:

0H	Drive A:
1H	Drive B:
2H	Drive C:
3H	Drive D:

## 26 Set DMA Address

### Purpose

Changes the default DMA address.

### Parameters

Parameter	Register	Contents
Entry point	C	1AH
Entry value	DE	DMA address
Returned value	None	None

### Remarks

Set DMA Address changes the default DMA address. DMA (Direct Memory Address) is a method of transferring data directly between memory and the disk subsystem. In CP/M, the DMA address is the address of the 128-byte data record before a disk write operation, or after a disk read operation occurs.

When a cold start, warm start, or disk system reset operation is performed, the DMA address automatically resets to 0080H. The DMA address can be changed with the Set DMA Address call to access another area of memory where data records reside. The DMA address specified in register pair DE remains unchanged until the next Set DMA Address call, cold start, warm start, or disk system reset operation is performed.

## 27 Get Addr Alloc

### Purpose

Returns the allocation vector base address of the active drive.

### Parameters

Parameter	Register	Contents
Entry point	C	1BH
Entry value	None	None
Returned value	HL	Allocation vector address

### Remarks

CP/M maintains an allocation vector in memory for each on-line disk drive. Programs such as STAT and PIP use the information provided by the allocation vector to determine the amount of remaining storage.

### Note

Allocation vector information can be invalid if the selected drive has a read only attribute. Get Addr Alloc is not normally used by application programs.

## 28 Write Protect Disk

### Purpose

Sets write-protect status on the active drive.

### Parameters

Parameter	Register	Contents
Entry point	C	1CH
Entry value	None	None
Returned value	None	None

### Remarks

Write Protect Disk sets a temporary write-protect attribute on the active drive which disables write operations. The attribute is removed by the next cold or warm start.

## 29 Get Read/Only Vector

### Purpose

Determines which drives have the temporary read/only bit set.

### Parameters

Parameter	Register	Contents
Entry point	C	1DH
Entry value	None	None
Returned value	HL	R/O vector value

### Remarks

Get Read/Only Vector determines which drives have the temporary read/only bit set through a 16-bit vector in register HL. The least significant bit position in register L denotes the first drive (A:), and the most significant bit position in register H denotes the sixteenth drive (P:). A "0" bit indicates that the drive is R/W, while a "1" bit indicates the drive is R/O. The R/O bit is set either by system call 28, Write Protect Disk, or automatically by CP/M when it detects a changed disk.

## 30 Set File Attributes

### Purpose

Sets file attributes from a program.

### Parameters

Parameter	Register	Contents
Entry point	C	1EH
Entry value	DE	FCB address
Returned value	A	Directory code

### Remarks

Set File Attributes allows a program to change attributes of the file specified by the addressed FCB. Specifically, this system call either sets or resets the read only and system attributes in the FCB t1—t2 field. When called, Set File Attributes searches for a matching FCB, and changes the matched directory entry to contain the selected attributes.

### Note

Although the FCB indicators f1' through f4' are not currently used, they can be useful for application programs. (f1 and f4 are not involved in the matching process during file open and close operations.) Indicators f5' through f8' and t3' are reserved for future system expansion.

## 31 Get Addr Disk Parm

### Purpose

Reads the address of the disk parameters into register A.

### Parameters

Parameter	Register	Contents
Entry point	C	1FH
Entry value	None	None
Returned value	A	DPB address

### Remarks

Get Addr Disk Parm returns the address of the BIOS disk parameter block in register pair HL. This address can be used for the following purposes:

1. To get the disk parameter values for display
2. To compute the amount of free disk space
3. To change the current disk parameter values

Normally, application programs will not require the use of this system call.

## 32 Set/Get User Code

### Purpose

Reads or changes the current user code.

### Parameters

Parameter	Register	Contents
Entry point	C	20H
Entry value	E	0FFH (get) or user code (set)
Returned value	A	Current code or 0FFH (no value)

### Remarks

Set/Get User Code allows an application program to read or change the current user number. To read the current user number, register E must contain the value 0FFH. Set/Get User Code will return the value of the current user number (0 to 15) in register A. If the value in register E is not 0FFH, then the current number is changed to the value of E (modulo 32).



### 33 Read Random

#### Purpose

Reads a record using random (direct) access.

#### Parameters

Parameter	Register	Contents
Entry point	C	21H
Entry value	DE	FCB address
Returned value	A	Return code

#### Remarks

Read Random is similar to system call 20, Read Sequential, except that the read operation takes place at the record number selected by the r0—r2 field of the FCB. Read operations only use bytes r0 and r1. (Byte r2 is used only in computing the size of a file. See system call 35, Compute File Size, for more information on computing the size of a file.)

The r0—r1 byte pair contains the value corresponding to the record to be read. The value range of r0—r1 (0H to 65535H) can access any particular record of an eight-megabyte file. Byte r2 must be set to zero, since a non-zero value indicates overflow past the end of the file.

Before a file can be read with a Random Read call, it must be opened with either an Open File or Make File system call. This ensures that the information in the file's FCB is read into the FCB contained in the DE register pair. When the file is opened, the selected record number is read into the FCB record field (r0—r1), and then Read Random can read the record. When the call is completed, register A contains either the value 00H to indicate a successful read operation, or an error code.

When the read operation has been completed, the current DMA buffer will contain the data of the selected record.

The FCB record number is not incremented by the system call. This differs from a Read Sequential system call where the record number is incremented. By not incrementing the record number, subsequent Read Random system calls continue to read the same record.

After each Read Random call, the logical extent and current record values are automatically set to the appropriate values to allow the file to be sequentially read or written, starting from the current randomly accessed position. The random record position can be advanced optionally by the program following each random read or write operation to obtain the effect of a sequential I/O operation.

---

### *Note*

The first Read Sequential call after a Read Random call rereads the record in the DMA buffer.

---

The following error codes are returned in register A, if the read operation was unsuccessful:

01H	Reading unwritten data
03H	Cannot close current extent
04H	Seek to unwritten extent
06H	Seek past physical end-of-disk

Error codes 01H and 04H occur when a random read operation accesses a data block which has not been previously written, or an extent which has not been created. These are equivalent conditions. Error code 03H does not normally occur under proper system operation, but can be cleared by simply rereading, or reopening extent zero as long as the disk is not physically write-protected. Error code 06H occurs whenever byte r2 contains a non-zero value under the current CP/M version 2.0 release. Normally, non-zero return codes can be treated as missing data, with zero return codes indicating that the operation is complete.

## 34 Write Random

### Purpose

Writes a record using random (direct) access.

### Parameters

Parameter	Register	Contents
Entry point	C	22H
Entry value	DE	FCB
Returned value	A	Return code

### Remarks

The Write Random system call is similar to the Read Random system call, except that data is written to the specified file on disk from the current DMA buffer.

If the addressed file's extent has not yet been allocated, the allocation is performed before the write operation continues. As in the Read Random system call, the random record number is not changed as a result of the write operation. The extent number and current record fields of the addressed FCB are set to correspond to the random record which is being written.

After a random write operation has been performed, sequential read or write operations can commence with the notation that the currently addressed record is to be either read or written again as the sequential operation begins. The random record field can also be advanced by the programmer following each write operation to achieve the effect of a sequential write operation.

---

***Note***

Reading or writing the last record of an extent in random mode does not cause an automatic extent switch as it does in sequential mode.

---

The error codes returned by a random write are identical to the random read operation, with the addition of error code 05H, which indicates that a new extent cannot be created due to directory overflow.

## 35 Compute File Size

### Purpose

Determines the size of the file.

### Parameters

Parameter	Register	Contents
Entry point	C	23H
Entry value	DE	FCB address
Returned value	FCB	Random record field set

### Remarks

Compute File Size determines the size of the file specified in the DE register pair. The FCB in register pair DE cannot contain wild card characters and the r0—r2 field is used for random access.

When the call is completed, the r0—r2 field of the FCB contains the record address of the virtual file size. If the value r2 is 01H, the file contains 65536 records, which is the maximum size of a file. If r2 is 00H, r0 and r1 contain the file size, which is a 16-byte value with r0 as the least significant byte.

Compute File Size can be used to append data to the end of a file by setting the random record position to the end of the file and then performing a sequence of random write operations, starting at the preset record address.

---

### ***Note***

If the file is written to by sequential write operations, the virtual size of a file is the same as the physical size. If the file was written to in random mode, "holes" exist in the allocation map and the file may contain fewer records than the size indicates. For example, if only the last record of an eight-megabyte file is written in random mode (i.e., record number 65535), then the virtual size is 65536 records, although only one block of data is actually allocated.

---

## 36 Set Random Record

### Purpose

Sets the random record position from a sequentially accessed file to a specific value.

### Parameters

Parameter	Register	Contents
Entry point	C	24H
Entry value	DE	FCB address
Returned value	None	Random record field set

### Remarks

Set Random Record sets the random record field of the specified file to a new value. This system call can be used in two ways.

First, it can eliminate the task of searching a sequentially accessed file to get the contents of various "key" fields. As each field is encountered, Set Random Record is called to compute the random record position for the data corresponding to this key. If the size of the data block is 128 bytes, the record position is placed into a table with the key for later retrieval.

After searching the entire file and tabulating the key fields and their record numbers, you can move instantly to a particular keyed record by performing a random read operation and by using the corresponding random record number which was saved earlier. This method can be used when variable record lengths are involved, since the program need only store the buffer-relative byte position along with the key field and record number to find the exact starting position of the keyed data.

The second use of Set Random Record is for switching from sequential access operations to random access operations. If a file is sequentially accessed to a particular point in the file, Set Random Record is called to set the record number. Subsequent random read and write operations continue from the selected point in the file.



## 37 Reset Drive

### Purpose

Resets specified disk drives to their initial values.

### Parameters

Parameter	Register	Contents
Entry point	C	25H
Entry value	DE	Drive vector
Returned value	A	00H

### Remarks

Reset Drive allows a calling program to reset a specified drive. The drive vector parameter is a 16-bit vector of the drive to be reset where the least significant bit represents drive A:.

## 40 Write Random With Zero Fill

### Purpose

Writes a zero record using random (direct) access.

### Parameters

Parameter	Register	Contents
Entry point	C	28H
Entry value	DE	FCB address
Returned value	A	Return code

### Remarks

Write Random With Zero Fill is similar to system call 34, Write Random, but writes zeros into a previously unallocated block before data is written.