

TABLA DE SERVICIOS DE INTERRUPCION DEL MS DOS.

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Program terminate				
Program Terminate	20H	CS = address of PSP		<ul style="list-style-type: none"> - INT 20 restores the terminate, Ctrl-Break, and critical error exit addresses from values in the PSP, flushes all buffers, frees memory and returns to DOS via the termination handler address - does not close FCBs - this function is no longer recommended, but can be used by version of DOS before 2.0, see INT 21,4C and INT 21,0
DOS Function Dispatcher				
Program Terminate	21H	AH = 00 CS = PSP segment address		<ul style="list-style-type: none"> - restores the terminate, Ctrl-Break, and critical error exit addresses, flushes all buffers, frees memory and returns to DOS via the termination handler address - does not close FCBs - this function is no longer recommended, but can be used by version of DOS before 2.0, see INT 21,4C and INT 20
Keyboard Input with Echo	21H	AH = 01	AL = character from standard input device	<ul style="list-style-type: none"> - waits for keyboard input from STDIN and echoes to STDOUT - returns 0 for extended keystroke, then function must be called again to return scan code - if Ctrl-Break is detected, INT 23 is executed
Display Output	21H	AH = 02 DL = character to output		<ul style="list-style-type: none"> - outputs character to STDOUT - backspace is treated as non-destructive cursor left- if Ctrl-Break is detected, INT 23 is executed
Wait for Auxiliary Device Input	21H	AH = 03	AL = character from the auxiliary device	<ul style="list-style-type: none"> - does not supply error returns - waits for character and reads from STDAUX - default DOS AUX parameters are 2400,N,8,1
Auxiliary Output	21H	AH = 04 DL = character to output		<ul style="list-style-type: none"> - sends character in DL to STDAUX - does not supply error returns - waits until STDAUX is available - default DOS AUX parameters are 2400,N,8,

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Printer Output	21H	AH = 05 DL = character to output		- sends character in DL to STDPRN - waits until STDPRN device is ready before output
Direct Console I/O	21H	AH = 06 DL = (0-FE) character to output = FF if console input request	AL = input character if console input request (DL=FF) ZF = 0 if console request character available (in AL) = 1 if no character is ready, and function request was console input	- reads from or writes to the console device depending on the value of DL - cannot output character FF (DL=FF indicates read function) - for console read, no echo is produced - returns 0 for extended keystroke, then function must be called again to return scan code - ignores Ctrl-Break and Ctrl-PrtSc
Direct Console Input Without Echo	21H	AH = 07	AL = character from STDIN	- waits for keyboard input until keystroke is ready - character is not echoed to STDOUT - returns 0 for extended keystroke, then function must be called again to return scan code - ignores Ctrl-Break and Ctrl-PrtSc - see INT 21,1
Console Input Without Echo	21H	AH = 08	AL = character from STDIN	- returns 0 for extended keystroke, then function must be called again to return scan code - waits for character from STDIN and returns data in AL - if Ctrl-Break is detected, INT 23 is executed
Print String	21H	AH = 09 DS:DX = pointer to string ending in "\$"		- outputs character string to STDOUT up to "\$" - backspace is treated as non-destructive - if Ctrl-Break is detected, INT 23 is executed
Buffered Keyboard Input	21H	AH = 0A DS:DX = pointer to input buffer		Pointer format: max count BUFFER (N bytes) `----- input buffer `----- number of characters returned (byte) `----- maximum number of characters to read (byte) - since strings can be pre-loaded, it is recommended that the default string be terminated with a CR - N bytes of data are read from STDIN into buffer+2 - max buffer size is 255, minimum buffer size is 1 byte - chars up to and including a CR are placed into the buffer beginning at byte 2; Byte 1 returns the number of chars placed into the buffer (extended codes take 2 characters) - DOS editing keys are active during this call - INT 23 is called if Ctrl-Break or Ctrl-C detected

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Check Standard Input Status	21H	AH = 0B	AL = 00 if no character available = FF if character available	- checks STDIN for available characters - character is not returned - if Ctrl-Break is detected INT 23 is executed
Clear Keyboard Buffer and Invoke Keyboard Function	21H	AH = 0C AL = 01, 06, 07, 08 or 0A (INT 21 input functions)		see return values from INT 21,AL where AL is 1, 6, 7, 8 or A - main function is to clear the input buffer and call INT 21h with the specified function (in AL) - see INT 21,1, INT 21,6, INT 21,7, INT 21,8 & INT 21,A
Disk Reset	21H	AH = 0D		- all file buffers are flushed to disk - does NOT update directory entry
Select Disk	21H	AH = 0E DL = zero based, drive number 0-25, A: - Z:)	AL = one based, total number of logical drives including hardfiles (1-26)	- for DOS 3.x+, this function returns the number of logical drives or the value of LASTDRIVE (default of 5) in the CONFIG.SYS file
Open file	21H	AH = 0FH DS:DX = FCB	AL = result code	Obsolete. Use function 3DH instead
Close a File Using FCB	21H	AH = 10h DS:DX = pointer to opened FCB	AL = 00 if file closed = FF if file not closed	Obsolete. Use function 3EH instead
Search for First Entry Using FCB	21H	AH = 11h DS:DX = pointer to unopened FCB	AL = 00 if matching file found = FF if file not found	Obsolete. Use function 4EH instead
Search for Next Entry Using FCB	21H	AH = 12h DS:DX = pointer to unopened FCB returned from INT 21,11 or INT 21,12	AL = 00 if file found = FF if file not found	Obsolete. Use function 4FH instead

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Delete File Using FCB	21H	AH = 13h DS:DX = pointer to an unopened FCB	AL = 00 if file deleted = FF if file not found	Obsolete. Use function 41H instead
Sequential Read Using FCB	21H	AH = 14h DS:DX = pointer to an opened FCB	AL = 00 if successful read = 01 if end of file (no data read) = 02 if DTA is too small = 03 if end of file or partial record read	Obsolete. Use function 3FH instead
Sequential Write Using FCB	21H	AH = 15h DS:DX = pointer to an opened FCB	AL = 00 if write was successful = 01 if diskette is full or read only = 02 if DTA is too small	Obsolete. Use function 40H instead
Create a File Using FCB	21H	AH = 16h DS:DX = pointer to an unopened FCB	AL = 00 if file Created = FF if file creation failed	Obsolete. Use function 3CH, 5AH or 5BH instead
Rename a File Using FCB	21H	AH = 17h DS:DX = pointer to a modified FCB of the format: Offset Description 00 drive designator 01 original file name 09 original file ext. 11 new file name 19 new extension	AL = 00 if file renamed = FF if file not renamed	Obsolete. Use function 56H instead
Get Current Default Drive	21H	AH = 19h	AL = current default drive (0=A,1=B,etc)	- determines the current default drive

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Set Disk Transfer Address (DTA)	21H	AH = 1A DS:DX = pointer to disk transfer address (DTA)		- specifies the disk transfer address to DOS - DTA cannot overlap 64K segment boundary - offset 80h in the PSP is a 128 byte default DTA supplied by DOS upon program load - use of the DTA provided by DOS will result in the loss of the program command tail which also occupies the 128 bytes starting at offset 80h of the PSP - see INT 21,2F
Get Allocation Table Information	21H	AH = 1B	AL = sectors per cluster CX = bytes per sector DX = clusters on disk DS:BX = pointer to Media Descriptor Byte found in FAT	Obsolete. Use function 36H instead
Get Allocation Table Info for Specified Drive	21H	AH = 1C DL = drive number (0 for default, 1 = A:, Z: = 26)	AL = sectors per cluster CX = bytes per sector DX = clusters on disk DS:BX = pointer to Media Descriptor Byte found in FAT	Obsolete. Use function 36H instead
Random Read Using FCB	21H	AH = 21h DS:DX = pointer to an opened FCB	AL = 00 if read successful = 01 if EOF (no data read) = 02 if DTA is too small = 03 if EOF (partial record read)	Obsolete. Use function 3FH instead
Random Write Using FCB	21H	AH = 22h DS:DX = far pointer to an opened FCB	AL = 00 if write Successful = 01 if diskette full or read only = 02 if DTA is too small	Obsolete. Use function 40H instead

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Get File Size Using FCB	21H	AH = 23h DS:DX = pointer to an unopened FCB	AL = 00 if successful = FF if file not found	Obsolete. Use function 42H instead
Set Relative Record Field in FCB	21H	AH = 24h DS:DX = pointer to an opened FCB		Obsolete. Use function 42H instead
Set Interrupt Vector	21H	AH = 25h AL = interrupt number DS:DX = pointer to interrupt handler		- provides a safe method for changing interrupt vectors - see INT 21,35
Create New Program Segment Prefix	21H	AH = 26h DX = segment address of new PSP		Obsolete. Use function 4BH instead
Random Block Read Using FCB	21H	AH = 27h CX = number of records to read DS:DX = pointer to an opened FCB	AL = 00 if read was successful = 01 if EOF (no data read) = 02 if DTA is too small = 03 if EOF (partial record read) CX = actual number of records read	Obsolete. Use function 3FH instead
Random Block Write Using FCB	21H	AH = 28h CX = number of records to write DS:DX = pointer to an opened FCB	AL = 00 if write successful = 01 if diskette full or read only = 02 if DTA is too small CX = number of records written	Obsolete. Use function 40H instead

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Parse a Filename for FCB	21H	AH = 29h AL = bit pattern to control parsing DS:SI = pointer to a filespec to parse ES:DI = pointer to a buffer for unopened FCB	AL = 00 if no wildcard characters present = 01 if wildcards present in string = FF if drive specifier is invalid DS:SI = pointer to the first character after parsed filename ES:DI = pointer to the updated unopened FCB	Bit patterns for parsing control found in AL: 7 6 5 4 3 2 1 0 AL `---- 1 = ignore leading separators 0 = don't ignore leading separators `----- 1 = modify drive ID if specified 0 = modify drive ID regardless `----- 1 = modify filename if specified 0 = modify filename regardless `----- 1 = modify extension if specified 0 = modify extension regardless `----- unused - retrieves filename from the command line string and places the filename components into an unopened FCB for later use - if no filename is found a pointer is returned in ES:DI that has a blank at ES:DI+1 - this function can be used to detect the existence of logical DOS drives by creating a dummy filespec with a drive letter and colon prepended. If the drive is invalid, this function will return FF in AL
Get Date	21H	AH = 2AH	AL = day of the week (0=Sunday) CX = year (1980-2099) DH = month (1-12) DL = day (1-31)	- retrieves system date based on the DOS maintained clock - updates BIOS Data Area current date and date rollover flag at location 40:70
Set Date	21H	AH = 2BH CX = year (1980-2099) DH = month (1-12) DL = day (1-31)	AL = 00 if date change successful = FF if invalid date	- sets DOS maintained clock - DOS versions 3.3+ also update the CMOS date where applicable
Get Time	21H	AH = 2C	CH = hour (0-23) CL = minutes (0-59) DH = seconds (0-59) DL = hundredths (0-99)	- retrieves DOS maintained clock time
Set Time	21H	AH = 2D CH = hour (0-23) CL = minutes (0-59) DH = seconds (0-59) DL = hundredths (0-99)	AL = 00 if time change successful = FF if time invalid	- changes DOS maintained clock - DOS version 3.3+ also update CMOS clock where applicable

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Set/Reset Verify Switch	21H	AH = 2E AL = 00 to set off = 01 to set verify on DH = 00 for DOS versions before 3.0		- with the verify setting on, disk I/O is more secure but takes longer to complete - see INT 21,54
Get Disk Transfer Address (DTA)	21H	AH = 2F	ES:BX = pointer to current DTA	- returns the DTA address - the default DTA is a 128 byte block at address 80h in the Program Segment Prefix (PSP). This area also contains the command tail at program startup it must be saved or the DTA must be relocated before use to preserve the command tail - see INT 21,1A
Get DOS Version Number	21H	AH = 30h	AL = major version number (2-5) AH = minor version number (in hundredths decimal) BH = FF indicates MS-DOS, only if OEM vendor chooses to identify = 00 indicates PC-DOS BL:CX = 24 bit OEM serial number if BH is FF	- for an example DOS version 2.1 returns AL=2 and AH=10 - DOS versions prior to DOS 2.0 return zero in AH and AL - DOS version 4.0 and 4.1 usually return the same value of 4.00 - the OEM serial number is a rarity, though some older OEM DOS versions implemented this feature - the OS/2 compatibility box returns 10.10 for OS/2 1.1, 10.20 for OS/2 1.2, etc... - when testing for version, a specific test can often cause your code to not work in following versions of DOS. It is often better to test for a version number greater or equal to the minimum rather than a specific version number where possible
Terminate Process and Remain Resident	21H	AH = 31h AL = exit code (returned to batch files) DX = memory size in paragraphs to reserve		- preferred method for Terminate and Stay Resident programs - terminates process without releasing allocated memory and without closing open files - attempts allocation of memory specified in DX from memory allocated by DOS at startup. INT 21,48 memory allocation is not affected - see INT 27

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Get/Set System Values (Ctl-Break/Boot Drive)	21H	AH = 33h AL = 00 to get Ctrl-Break checking flag = 01 to set Ctrl-Break checking flag = 02 to set extended Ctrl-Break checking = 05 get boot drive (DOS 4.x) DL = 00 to set Ctrl-Break checking off = 01 to set Ctrl-Break checking on = boot drive for subfunction 5; (1=A:, 2=B:, ...)	DL = 00 Ctrl-Break checking OFF (AL=0 or AL=2) = 01 Ctrl-Break checking ON (AL=0 or AL=2) = boot drive number (1-26, A: - Z:) (function 05)	- retrieves DOS Ctrl-Break or extended Ctrl-Break setting which determines if DOS will check for Ctrl-Break during INT 21 calls
Get Interrupt Vector	21H	AH = 35h AL = interrupt vector number	ES:BX = pointer to interrupt handler	- standard method for retrieving interrupt vectors - see INT 21,25
Get Disk Free Space	21H	AH = 36h DL = drive number (0=default, 1=A:)	AX = sectors per cluster = FFFF if drive is invalid BX = number of available clusters CX = number of bytes per sector DX = number of clusters per drive	- used to determine available space on specified disk - see INT 21,1B INT 21,1C

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Get/Set Country Dependent Information	21H	AH = 38h AL = 00 to get current country information = 00-FE country codes (DOS 3.x+) = FF for country codes >= FF, country codes are in BX BX = country code if AL = FF (country code > 255) DX = FFFF to set country information DS:DX = pointer to buffer to contain country data (if get data)	AX = error code if CF set = 02 invalid country BX = country code (DOS 3.x+) DS:DX = pointer to returned country data	- returns a pointer to country specific data, for DOS 3.x+ this function can be used to also set this information
Create Subdirectory (mkdir)	21H	AH = 39h DS:DX = pointer to ASCII path name	CF = 0 if successful = 1 if error AX = error code (see DOS ERROR CODES)	- creates specified subdirectory - returns error if directory already exists, element of the path is not found, directory full or write protected disk
Remove Subdirectory (rmdir)	21H	AH = 3Ah DS:DX = pointer to ASCII path name	CF = 0 if successful = 1 if error AX = error code (see DOS ERROR CODES)	- allows deletion of a subdirectory as long as it exists, is empty and not the current directory
Change Current Directory (chdir)	21H	AH = 3Bh DS:DX = pointer to ASCII path name	CF = 0 if successful = 1 if error AX = error code if CF set (see DOS ERROR CODES)	- changes the current directory to the directory specified by pointer DS:DX
Create File Using Handle	21H	AH = 3Ch CX = file attribute (see FILE ATTRIBUTES) DS:DX = pointer to ASCII path name	CF = 0 if successful = 1 if error AX = files handle if successful = error code if failure (see DOS ERROR CODES)	- if file already exists, it is truncated to zero bytes on opening

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Open File Using Handle	21H	AH = 3DH AL = open access mode 00 read only 01 write only 02 read/write DS:DX = pointer to an ASCIIZ file name	AX = file handle if CF not set = error code if CF set	Access modes in AL: 7 6 5 4 3 2 1 0 AL `----- read/write/update access mode `----- reserved, always 0 `----- sharing mode (see below) (DOS 3.1+) `----- 1 = private, 0 = inheritable (DOS 3.1+) Sharing mode bits (DOS 3.1+): Access mode bits: 654 210 000 compatibility mode (exclusive) 000 read access 001 deny others read/write access 001 write access 010 deny others write access 010 read/write access 011 deny others read access 100 full access permitted to all - will open normal, hidden and system files - file pointer is placed at beginning of file
Close File Using Handle	21H	AH = 3EH BX = file handle to close	AX = error code if CF set	- if file is opened for update, file time and date stamp as well as file size are updated in the directory - handle is freed
Read From File or Device Using Handle	21H	AH = 3FH BX = file handle CX = number of bytes to read DS:DX = pointer to read buffer	AX = number of bytes read is CF not set = error code if CF set	- read specified number of bytes from file into buffer DS:DX - when AX is not equal to CX then a partial read occurred due to end of file - if AX is zero, no data was read, and EOF occurred before read
Write To File or Device Using Handle	21H	AH = 40h BX = file handle CX = number of bytes to write, a zero value truncates/extends the file to the current file position DS:DX = pointer to write buffer	AX = number of bytes written if CF not set = error code if CF set	- if AX is not equal to CX on return, a partial write occurred - this function can be used to truncate a file to the current file position by writing zero bytes

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Delete File	21H	AH = 41h DS:DX = pointer to an ASCIIIZ filename	AX = error code if CF set	- marks first byte of file directory entry with E5 to indicate the file has been deleted. The rest of the directory entry stays intact until reused. FAT pointers are returned to DOS - documented as not accepting wildcards in filename but actually does in several DOS versions
Move File Pointer Using Handle	21H	AH = 42h AL = origin of move: 00 = beginning of file plus offset (SEEK_SET) 01 = current location plus offset (SEEK_CUR) 02 = end of file plus offset (SEEK_END) BX = file handle CX = high order word of number of bytes to move DX = low order word of number of bytes to move	AX = error code if CF set DX:AX = new pointer location if CF not set	- seeks to specified location in file
Get/Set File Attributes	21H	AH = 43h AL = 00 to get attribute = 01 to set attribute DS:DX = pointer to an ASCIIIZ path name CX = attribute to set	AX = error code if CF set CX = the attribute if AL was 00	attribute to set: 5 4 3 2 1 0 CX valid file attributes `---- 1 = read only `----- 1 = hidden `----- 1 = system `----- not used for this call `----- 1 = archive

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
I/O Control for Devices (IOCTL)	21H	AH = 44h AL = function value BX = file handle BL = logical device number (0=default, 1=A:, 2=B:, 3=C:, ...) CX = number of bytes to read or write DS:DX = data or buffer	AX = error code if CF set AX = # of bytes transferred if CF not set	Functions (AL value): IOCTL,0 Get Device Information IOCTL,1 Set Device Information IOCTL,2 Read From Character Device IOCTL,3 Write to Character Device IOCTL,4 Read From Block Device IOCTL,5 Write to Block Device IOCTL,6 Get Input Status IOCTL,7 Get Output Status IOCTL,8 Device Removable Query IOCTL,9 Device Local or Remote Query IOCTL,A Handle Local or Remote Query IOCTL,B Set Sharing Retry Count IOCTL,C Generic I/O for Handles IOCTL,D Generic I/O for Block Devices (3.2+) IOCTL,E Get Logical Drive (3.2+) IOCTL,F Set Logical Drive (3.2+)
Duplicate File Handle	21H	AH = 45h BX = file handle	AX = new file handle if CF not set = error code if CF set	- gets another file handle for the same file - both file handles move in unison - often used to flush file data and update a file directory entry without closing the initial file
Force Duplicate File Handle	21H	AH = 46h BX = existing file handle CX = second file handle	AX = error code if CF set	- if file handle in CX is currently open, current file identified by CX is closed and the handle in BX is dupped and placed in CX - after duping file handles move together through file - similar to INT 21,45
Get Current Directory	21H	AH = 47h DL = drive number (0 = default, 1 = A:) DS:SI = pointer to a 64 byte user buffer	DS:SI = pointer ASCIIZ directory path string AX = error code if CF set	- returns the current directory relative to the root directory - the leading slash "\" and drive designator are omitted

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Allocate Memory	21H	AH = 48h BX = number of memory paragraphs requested	AX = segment address of allocated memory block (MCB + 1para) = error code if CF set BX = size in paras of the largest block of memory available if CF set, and AX = 08 (Not Enough Mem) CF = 0 if successful = 1 if error	- returns segment address of allocated memory block AX:0000 - each allocation requires a 16 byte overhead for the MCB - returns maximum block size available if error - see INT 21,49, INT 21,4A
Free Allocated Memory	21H	AH = 49h ES = segment of the block to be returned (MCB + 1para)	AX = error code if CF set	- releases memory and MCB allocated by INT 21,48 - may cause unpredictable results is memory wasn't allocated using INT 21,48 or if memory wasn't allocated by the current process - checks for valid MCB id, but does NOT check for process ownership - care must be taken when freeing the memory of another process, to assure the segment isn't in use by a TSR or ISR - this function is unreliable in a TSR once resident, since COMMAND.COM and many other .COM files take all available memory when they load -see INT 21,4A
Modify Allocated Memory Block (SETBLOCK)	21H	AH = 4Ah BX = new requested block size in paragraphs ES = segment of the block (MCB + 1 para)	AX = error code if CF set) BX = maximum block size possible, if CF set and AX = 8	- modifies memory blocks allocated by INT 21,48 - can be used by programs to shrink or increase the size of allocated memory - PC-DOS version 2.1 and DOS 3.x will actually allocate the largest available block if CF is set. BX will equal the size allocated. - see also INT 21,49

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS																											
		ENTRADA	SALIDA																												
EXEC/Load and Execute Program	21H	AH = 4BH AL = 00 to load and execute program = 01 (Undocumented) create program segment prefix and load program, but don't execute. The CS:IP and SS:SP of the program is placed in parameter block. Used by debuggers = 03 load program only = 04 called by MSC spawn() when P_NOWAIT is specified DS:DX = pointer to an ASCII filename ES:BX = pointer to a parameter block	AX = error code if CF set ES:BX = when AL=1, pointer to parameter block similar to:	<table border="0"> <thead> <tr> <th>Offset</th> <th>Size</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>word</td> <td>when AL=1, segment of env. or zero if using parents env.</td> </tr> <tr> <td></td> <td>word</td> <td>when AL=3, segment of load point for overlay</td> </tr> <tr> <td>02</td> <td>dword</td> <td>when AL=1, pointer to cmd line at PSP 80h</td> </tr> <tr> <td></td> <td>word</td> <td>when AL=3, relocation factor for EXE overlay</td> </tr> <tr> <td>06</td> <td>dword</td> <td>pointer to default FCB passed at PSP 5Ch</td> </tr> <tr> <td>0A</td> <td>dword</td> <td>pointer to default FCB passes at PSP 6Ch</td> </tr> <tr> <td>0E</td> <td>dword</td> <td>value of program SS:SP</td> </tr> <tr> <td>12</td> <td>dword</td> <td>value of program CS:IP</td> </tr> </tbody> </table> <ul style="list-style-type: none"> - allows execution of an external program as well as overlay management from within an application - all registers except CS and IP are destroyed - SS and SP should be preserved in code segment before call since a bug in DOS version 2.x destroys these - return code can be retrieved if child process exits via INT 21,4C - calling process must assure presence of enough unallocated memory - subfunction 4 returns with an error under DOS 4.x+ - calls INT 21,55 - see also INT 21,26 	Offset	Size	Description	00	word	when AL=1, segment of env. or zero if using parents env.		word	when AL=3, segment of load point for overlay	02	dword	when AL=1, pointer to cmd line at PSP 80h		word	when AL=3, relocation factor for EXE overlay	06	dword	pointer to default FCB passed at PSP 5Ch	0A	dword	pointer to default FCB passes at PSP 6Ch	0E	dword	value of program SS:SP	12	dword	value of program CS:IP
Offset	Size	Description																													
00	word	when AL=1, segment of env. or zero if using parents env.																													
	word	when AL=3, segment of load point for overlay																													
02	dword	when AL=1, pointer to cmd line at PSP 80h																													
	word	when AL=3, relocation factor for EXE overlay																													
06	dword	pointer to default FCB passed at PSP 5Ch																													
0A	dword	pointer to default FCB passes at PSP 6Ch																													
0E	dword	value of program SS:SP																													
12	dword	value of program CS:IP																													
Terminate Process With Return Code	21H	AH = 4Ch AL = return code (for batch files)		<ul style="list-style-type: none"> - approved method of program termination - restores the terminate, Ctrl-Break, and critical error exit addresses, flushes all buffers, frees memory and returns to DOS via the termination handler address - does not close FCBs - this function is not supported in versions of DOS before 2.x, so use INT 21,0 or INT 20 to exit. - see also INT 27 INT 21,31 																											

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS																																																			
		ENTRADA	SALIDA																																																				
Get Return Code of Sub-process	21H	AH = 4Dh	AH = system exit code (indicates normal termination) = 00 for normal termination = 01 if terminated by ctl-break = 02 if critical device error = 03 if terminated by INT 21,31 AL = child exit code	- retrieve child process and system exit codes - this function can only be used to retrieve the exit code once multiple attempts to read exit codes will cause errors																																																			
Find First Matching File	21H	AH = 4EH CX = attribute used during search DS:DX = pointer to ASCIIZ filespec, including wildcards	AX = error code if CF set DTA = data returned from call	Data returned from call in the format: <table border="1"> <thead> <tr> <th>Offset</th> <th>Size</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>byte</td> <td>attribute of search (undocumented)</td> </tr> <tr> <td></td> <td>byte</td> <td>drive letter used in search (DOS 3.1-4.x, undocumented)</td> </tr> <tr> <td>01</td> <td>byte</td> <td>drive letter used in search (undocumented)</td> </tr> <tr> <td></td> <td>11bytes</td> <td>search name used (DOS 3.1-4.x, undocumented)</td> </tr> <tr> <td>02</td> <td>11bytes</td> <td>search name used (undocumented)</td> </tr> <tr> <td>0C</td> <td>byte</td> <td>attribute of search (DOS 3.1-4.x, undocumented)</td> </tr> <tr> <td>0D</td> <td>word</td> <td>directory entry number (0 based, undocumented)</td> </tr> <tr> <td>0F</td> <td>word</td> <td>starting cluster number of current directory; zero for root directory (DOS 3.2+, undocumented)</td> </tr> <tr> <td></td> <td>dword</td> <td>pointer to DTA (DOS 2.x-3.1, undocumented)</td> </tr> <tr> <td>11</td> <td>word</td> <td>reserved</td> </tr> <tr> <td>13</td> <td>word</td> <td>starting cluster number of current directory; zero for root directory (DOS 2.x+, undocumented)</td> </tr> <tr> <td>15</td> <td>byte</td> <td>attribute of matching file</td> </tr> <tr> <td>16</td> <td>word</td> <td>file time</td> </tr> <tr> <td>18</td> <td>word</td> <td>file date</td> </tr> <tr> <td>1A</td> <td>word</td> <td>file size</td> </tr> <tr> <td>1E</td> <td>13bytes</td> <td>ASCIIZ filename and extension in the form NAME.EXT with blanks stripped</td> </tr> </tbody> </table> <p>- returns information on first file matching specifications - use INT 21,4F to retrieve following file matches - DOS 2.x cannot find . and .. entries, while DOS 3.x can unless they represent the root directory - character devices return a zero for size, time and</p>	Offset	Size	Description	00	byte	attribute of search (undocumented)		byte	drive letter used in search (DOS 3.1-4.x, undocumented)	01	byte	drive letter used in search (undocumented)		11bytes	search name used (DOS 3.1-4.x, undocumented)	02	11bytes	search name used (undocumented)	0C	byte	attribute of search (DOS 3.1-4.x, undocumented)	0D	word	directory entry number (0 based, undocumented)	0F	word	starting cluster number of current directory; zero for root directory (DOS 3.2+, undocumented)		dword	pointer to DTA (DOS 2.x-3.1, undocumented)	11	word	reserved	13	word	starting cluster number of current directory; zero for root directory (DOS 2.x+, undocumented)	15	byte	attribute of matching file	16	word	file time	18	word	file date	1A	word	file size	1E	13bytes	ASCIIZ filename and extension in the form NAME.EXT with blanks stripped
Offset	Size	Description																																																					
00	byte	attribute of search (undocumented)																																																					
	byte	drive letter used in search (DOS 3.1-4.x, undocumented)																																																					
01	byte	drive letter used in search (undocumented)																																																					
	11bytes	search name used (DOS 3.1-4.x, undocumented)																																																					
02	11bytes	search name used (undocumented)																																																					
0C	byte	attribute of search (DOS 3.1-4.x, undocumented)																																																					
0D	word	directory entry number (0 based, undocumented)																																																					
0F	word	starting cluster number of current directory; zero for root directory (DOS 3.2+, undocumented)																																																					
	dword	pointer to DTA (DOS 2.x-3.1, undocumented)																																																					
11	word	reserved																																																					
13	word	starting cluster number of current directory; zero for root directory (DOS 2.x+, undocumented)																																																					
15	byte	attribute of matching file																																																					
16	word	file time																																																					
18	word	file date																																																					
1A	word	file size																																																					
1E	13bytes	ASCIIZ filename and extension in the form NAME.EXT with blanks stripped																																																					

Tabla de servicios de Interrupcion del BIOS

				<p>date in DOS 2.x, while DOS 3.0 returns a 40h attribute and current time and date.</p> <ul style="list-style-type: none"> - multiple calls to this function with a character device will result in unpredictable results - normal files are always included along with files that match the requested attributes except when the LABEL attribute is requested. DOS 2.x returns all normal files when label is specified but 3.x doesn't. It's up to the programmer to determine which actually match the requested attributes. - bit 8 of CX (file attributes) indicates Novell Netware shareable - see INT 21,1A
--	--	--	--	---

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Find Next Matching File	21H	AH = 4FH DS:DX = unchanged from previous function 4E	AX = error code if CF set	<ul style="list-style-type: none"> - INT 21,4E should be called to find the first file and this function should be called to find all other matching files - normal files are always included along with files that match the requested attributes except when the LABEL attribute is requested. It's up to the programmer to determine which actually match the requested attributes. - see INT 21,1A
Set Current Process ID	21H	AH = 50h BX = process ID number (process PSP segment address)		<ul style="list-style-type: none"> - the process ID number is actually the segment address of a program's PSP. This is useful for TSR's to access their own file handle table inside their PSP. - this function cannot be called while in an INT 28 handler in DOS 2.x unless the critical error flag is set or stack corruption will occur - safe in DOS 3.x INT 28 handlers since they use a different stack by default - available in OS/2 compatibility box - see INT 21,51 INT 21,62
Get Current Process ID	21H	AH = 51h	BX = process ID	<ul style="list-style-type: none"> - The process ID number is actually the segment address of program's PSP. This in conjunction with INT 21,50 is useful for TSR's to access their own file handle table in their respective PSP. - this function cannot be called while in an INT 28 handler in DOS 2.x unless the critical error flag is set or stack corruption will occur - INT 21,62 is highly recommended for DOS 3.x due to a possible bug when activated from a TSR. DOS may switch to the wrong internal stack which may cause a problems with TSR's if called during an INT 28. - see INT 21,62 (Get PSP segment) for DOS 3.x applications
Get Verify Setting	21H	AH = 54h	AL = 00 verify off = 01 verify on	<ul style="list-style-type: none"> - returns value of disk read after write verification flag - see INT 21,2E

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Rename File	21H	AH = 56h DS:DX = pointer to old ASCIIZ path/filename ES:DI = pointer to new ASCIIZ path/filename	AX = error code if CF set	- supports full pathnames and allows renaming files across directories and in DOS 3.x allows renaming subdirectories - does not support use of wildcards unless invoked from via INT 21,5D in which case error code 12h is returned - unpredictable result may occur if an opened file is renamed
Get/Set File Date and Time Using Handle	21H	AH = 57h AL = 00 get date and time = 01 set date and time = 02 ??? (DOS 4.0+ undocumented) = 03 ??? (DOS 4.0+ undocumented) = 04 ??? (DOS 4.0+ undocumented) BX = file handle CX = time to set (if setting) DX = date to set (if setting) ES:DI = pointer to buffer to contain results	AX = error code if CF set) CX = file time (if reading, see note) DX = file date (if reading, see note)	Time encoding: F E D C B A 9 8 7 6 5 4 3 2 1 0 Time in CX `----- two second incr (0-29) `----- minutes 0-59) `----- hours (0-29) Date Encoding F E D C B A 9 8 7 6 5 4 3 2 1 0 Date in DX `----- day (1-31) `----- month (1-12) `----- year - 1980
Create Temporary File	21H	AH = 5A CX = attribute DS:DX = pointer to ASCIIZ path ending in '\\'	AX = handle if call CF clear = error code if CF set DS:DX = pointer to updated ASCIIZ filespec	- creates files with random names with any combination of the following attributes: normal, system and hidden - ending backslash is required
Create File	21H	AH = 5B CX = attribute DS:DX = pointer to ASCIIZ path/filename	AX = handle if CF not set = error code if CF set	- standard method of opening files - returns a file handle of a file opened with specified attributes (combinations of normal, system and hidden)

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Lock/Unlock File Access	21H	AH = 5C AL = 00 lock file = 01 unlock file BX = file handle CX = most significant word of region offset DX = least significant word of region offset SI = most significant word of region length DI = least significant word of region length	AX = error code if CF set	- used for networking and multi-tasking system to preserve data integrity - duplicated handles inherit access to locked regions - EXEC'd programs do not inherit access to locked regions
Get Machine Name	21H	AH = 5E AL = 00 get machine name DS:DX = far pointer to buffer receiving name string	AX = error code if CF set, (invalid function) CH = 0 if name not defined > 0 if name defined CL = NETBIOS name number, if CH not 0 DS:DX = far pointer to buffer containing string if CH not 0	- returns pointer to an ASCIIZ string identifying the computer on a Microsoft network - output string is a 15 bytes long, padded with blanks and null terminated - unpredictable if called without file sharing loaded
Set Machine Name	21H	AH = 5E AL = 01 set machine name CH = 00 undefine name = any other value means to define name CL = name number DS:DX = pointer to 15 byte, blank padded ASCIIZ name string	AX = error code if CF set, (invalid function)	- specifies the network machine name on a Microsoft network - unpredictable if called without file sharing loaded

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Set Printer Setup	21H	AH = 5E AL = 02 set printer setup BX = redirection list index (see INT 21,5F AL=2) CX = length of setup string DS:SI = far pointer to printer setup string	AX = error code if CF set, (invalid function)	- see INT 21,5F subfunction 2 for information on obtaining the redirection list index - fails if file sharing is not loaded
Get Printer Setup	21H	AH = 5E AL = 03 get printer setup BX = redirection list index (see INT 21,5F AL=2) ES:DI = far pointer to buffer to receive setup string	AX = error code if CF set	- see INT 21,5F subfunction 2 for information on obtaining the redirection list index - fails if file sharing is not loaded
Set Printer Mode	21H	AH = 5E AL = 04 set printer setup BX = redirection list index (see INT 21,5F AL=2) DX = printer mode to set (bit zero is set for binary mode and clear for translated/text mode)	AX = error code if CF set	- see INT 21,5F subfunction 2 for information on obtaining the redirection list index - fails if file sharing is not loaded - issues INT 2F,11 and INT 2F,1F
Get Printer Mode	21H	AH = 5E AL = 05 set printer setup BX = redirection list index (see INT 21,5F AL=2)	AX = error code if CF set DX = printer mode where bit zero is set for binary mode and clear for translated mode	- see INT 21,5F subfunction 2 for information on obtaining the redirection list index - fails if file sharing is not loaded - issues INT 2F,11 and INT 2F,1F

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Get Redirection Mode	21h	AH = 5F AL = 00 BL = redirection type = 03 printer = 04 disk drive	AX = error code if CF is set BH = device redirection state = 00 redirection off = 01 redirection on	- determines if disk or printer redirection is enabled - fails if file sharing is not loaded
Set Redirection Mode	21h	AH = 5F AL = 01 BH = device redirection state = 00 turn redirection off = 01 turn redirection on BL = redirection type = 03 printer = 04 disk drive	AX = error code if CF is set	- enables or disables disk or printer redirection - local devices are used when redirection is off - fails if file sharing is not loaded
Get Redirection List Entry	21h	AH = 5F AL = 02 BX = redirection list index DS:SI = far pointer to a 16 byte buffer to hold device name ES:DI = far pointer to a 128 byte buffer to hold network name	AX = error code if CF is set = 1 invalid function code = 12 no more files BH = device status flag, bit 0 = 0 device valid bit 0 = 1 device invalid BL = device type = 3 printer = 4 drive CX = stored parameter value DS:SI = far pointer to ASCIIZ local device name ES:DI = far pointer to ASCIIZ network name	- value returned in CX is the value previously passed to INT 21,5F with subfunction 3 in register CX - registers DX and BP are destroyed - fails if file sharing is not loaded

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Redirect Device	21h	AH = 5F AL = 03 BL = device type = 3 if printer = 4 if drive CX = parameter to save for caller DS:SI = far pointer to ASCIIZ local device name ES:DI = far pointer to ASCIIZ network name followed by ASCIIZ password	AX = error code if CF is set = 1 invalid function code or invalid source or destination string format or source device already redirected = 3 path not found = 5 access denied = 8 insufficient memory	- parameter passed in CX can later be retrieved by calls to INT 21,5F subfunction 2 - local device name in DS:SI can be a drive designator (D:), a printer name (PRN or LPTx), or a null string - when a null string is followed by a password is used, DOS tries granting access to the network directory using the given password - fails if file sharing is not loaded
Cancel Device Redirection	21h	AH = 5F AL = 04 DS:SI = far pointer to ASCIIZ local device name	AX = error code if CF set = 01 function code invalid ASCIIZ string is not an existing source device = 0F redirection paused on server	- local device name in DS:SI can be a drive designator (D:), a printer name (PRN or LPTx), or a string beginning with 2 backslashes "\\ " - when two backslashes are used for the string the connection between the local machine and network directory terminates - fails if file sharing is not loaded
Get Redirection List Extended Entry	21h	AH = 5F AL = 05 BX = redirection list index DS:SI = pointer to buffer to receive ASCIIZ source device name ES:DI = pointer to buffer to receive ASCIIZ network path	AX = error code if CF set BH = device status flag (bit 0 clear if valid) BL = device type = 03 printer device = 04 disk device CX = stored user parameter value BP = NETBIOS local session number DS:SI = pointer to buffer containing ASCIIZ source device name ES:DI = pointer to buffer containing ASCIIZ network path	- retrieves the status, type, source and target of a given redirection - fails if file sharing is not loaded

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Get PSP address	21h	AH = 62h	BX = segment address of current process	- this is useful for TSR's to access their own file handles - for DOS 2.x use INT 21,51 - see also INT 21,50
Get/Set Global Code Page	21h	AH = 66h AL = 01 get global code page = 02 set global code page BX = active code page if setting DX = system code page if setting	AX = error code if CF set BX = active code page (for AL=1 only) DX = system code page (for AL=1 only)	- moves country data from COUNTRY.SYS to country memory table
Set Handle Count	21h	AH = 67h BX = new maximum open handles allowed	CF = 0 if successful = 1 if error AX = error code if CF is set	- gives program control of the number of files simultaneously open - if CX is less than the current number of open files the change will take effect when the number of open handles falls below the new limit - this function allows the application to use more than 20 files, up to the FILES=N limit - earlier copies of IBM DOS 3.3 sometimes incorrectly allocates memory (up to 64K) if an even number of handles is requested - only the first 20 files handles are copied to a child process regardless of the max number of files - it is possible to allocate more than 255 file handles but it is difficult to use more than 255 - see SFT FILE HANDLES
Flush Buffer Using Handle	21h	AH = 68h BX = file handle	AX = error code if CF set	- flushes DOS buffers to disk, does not update directory entry

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Extended Open/Create	21H	AH = 6C AL = 00 BX = open mode (see below) CX = file attribute (see FILE ATTRIBUTES, bits 15-8 are zero) DX = function control indicator (see below) DS:SI = ASCIIZ filespec	AX = handle if CF clear = error code if CF set CX = 01 file opened = 02 file created and opened = 03 file truncated and opened (replaced) CF = 0 if successful = 1 if error	<p>Open Mode in BX:</p> <pre> F E D C B A 9 8 7 6 5 4 3 2 1 0 BX (Open Mode) `---- read/write access mode `----- reserve (must be 0) `----- sharing mode `----- 1 = private, 0 = inheritable `----- reserved `----- 0 = call INT 24, 1 = ignore INT 24 `----- 0 = normal write, 1 = flush each write `----- reserved </pre> <p>Read/Write Access bits: Sharing Mode bits: 210 654 000 read access 000 compatibility mode 001 write access 001 read/write access 010 read/write access 010 write access denied 011 read access denied 100 full access</p> <p>Function Control Indicator in DX:</p> <pre> F E D C B A 9 8 7 6 5 4 3 2 1 0 DX (Function Control) `---- file existence behavior `----- creation behavior `----- reserved </pre> <p>File Existence Behavior bits: 3210 0000 if file exist, generate error, take no action 0001 if file exists, open file 0010 if file exists, truncate and open</p> <p>Creation Behavior bits: 7654 0000 if file does not exist, generate error, take no action 0001 if file does not exist, create</p>
Program Terminate	22H			<ul style="list-style-type: none"> - not an interrupt, but a vector to the terminate address - copied to PSP offset 0Ah during program load - do not execute this interrupt directly

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Control-Break Exit Address	23H			<ul style="list-style-type: none"> - not an interrupt but a pointer to a routine that is called when a DOS function detects a Ctrl-Break or Ctrl-C has been pressed resulting in a "Break condition" - a Break condition is detected if DOS's internal Break flag is found set by INT 1B or the next word in the BIOS keyboard buffer is a scan code representing one of the Break key combinations. If a matching scan code is found, the keyboard buffer head pointer is incremented by 2 (which effectively removes the ^C keycode), the ^C is printed and then INT 23 is called. - disabling the abort is easy, but to disable the ^C from appearing you must either not use any of the DOS I/O functions that check for Break or you must trap the Ctrl-C keypress from within INT 9. Another method is to scan the BIOS keyboard buffer and delete any Break key combinations. - Ctrl-Break empties the keyboard buffer by resetting the keyboard head and tail pointers then places a 0000h at the queue head. It then sets an internal "Break" flag. DOS subfunctions that check for Break see this and then issue INT 23h. DOS does not detect INT 1B using the keyboard buffer, but uses it's own internal flag. - Alt-Keypad-3 and Ctrl-2 also result in this interrupt - Ctrl-C places 2E03h in the BIOS keyboard buffer while Ctrl-2 places 0300h and Alt-Keypad-3 places 0003h; none of these key combinations empty the keyboard like Ctrl-Break but all result in a ^C being displayed; note that all three produce scan codes containing '03' - since DOS checks only the first word of the keyboard buffer Ctrl-C, Ctrl-2 and Alt-Keypad-3 are only detected if they are the first word in the buffer while Ctrl-Break is detected via the internal flag and takes effect as soon as it's detected. - do not execute this interrupt directly - see also INT 9

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Absolute Disk Read	25H	AL = logical drive number (0=A:, 1=B:, 2=C:, ...) CX = number of sectors to read = -1 if DOS 4.x (control block pointer is in DS:BX) DX = starting logical sector number DS:BX = pointer to data buffer = pointer to control block	AH = error code if CF set AL = BIOS error code if CF set	AH = error code if CF set: 01 bad command 02 bad address mark 03 write protect 04 sector not found 08 DMA failure 10 data error (bad CRC) 20 controller failed 40 seek failed 80 attachment failed to respond AL = BIOS error code if CF set 00 write protect error 01 unknown unit 02 drive not ready 03 unknown command 04 data error (bad CRC) 05 bad request structure length 06 seek error 07 unknown media type 08 sector not found 0A write fault 0B read fault 0C general failure
Absolute Disk Write	26H	AL = logical drive number (0=A:, 1=B:, 2=C:, ...) CX = number of sectors to write = -1 if DOS 4.x (control block pointer is in DS:BX) DX = starting logical sector number DS:BX = pointer to data buffer = pointer to control block	AH = error code if CF set AL = BIOS error code if CF set	AH = error code if CF set: 01 bad command 02 bad address mark 03 write protect 04 sector not found 08 DMA failure 10 data error (bad CRC) 20 controller failed 40 seek failed 80 attachment failed to respond AL = BIOS error code if CF set 00 write protect error 01 unknown unit 02 drive not ready 03 unknown command 04 data error (bad CRC) 05 bad request structure length 06 seek error 07 unknown media type 08 sector not found 0A write fault 0B read fault 0C general failure

Tabla de servicios de Interrupcion del BIOS

SERVICIO	# INT.	REGISTROS		NOTAS
		ENTRADA	SALIDA	
Terminate and Stay Resident	27H	DX = offset of last byte in program to remain resident plus 1 CS = segment of PSP		<ul style="list-style-type: none"> - terminates process without closing open files - memory beyond the offset in DX is freed - see INT 21,31
Execute Command Using Base Level Command Interpreter	2EH	DS:SI = pointer to command string to be executed		<ul style="list-style-type: none"> - causes the base level COMMAND.COM to execute a command as if it were typed from the keyboard - the transient portion of COMMAND.COM must be loaded if not resident - first byte of the string contains the string length, inclusive - command string must be terminated with a CR - all registers including SS & SP are destroyed; SS & SP should be preserved in your own code segment - can be used to modify parents environment with SET command - incompatible under some software multitasking systems - before returning to the parent program this interrupt will attempt to execute the "current" batch file (if the parent or any ancestors were invoked from a batch file. This can cause remaining lines in a batch file to execute before the parent program finishes and fragment memory