

**Autore:** Michele Naso

**Classe:** QUINTA INFORMATICA (5IA)

**Anno scolastico:** 2003/2004

**Scuola:** Itis Euganeo

## INTERRUPT 21H

### Dispensa

Elenco di chiamate ai servizi DOS più diffusi e relativi schemi di utilizzazione.



### Uso dell'interruzione INT 21h per utilizzare i servizi e le funzioni del MS-DOS

L'interruzione sincrona (trap) INT 21H, serve per chiedere, da parte dell'utente, i servizi e le funzioni erogati dal DOS

In tutte le chiamate il valore contenuto nel registro AH contiene il numero della funzione che deve essere eseguita.

Altri parametri possono essere richiesti in altri registri.

In tutti i casi il modello di programmazione, da realizzare, deve essere in accordo alle indicazioni di seguito fornite:

- Impostare AH con il numero della funzione desiderata
- Impostare il contenuto dei registri implicati nella chiamata
- Fare la chiamata, alla funzione impostata, con l'istruzione INT 21h

Schema di chiamata

Esempio: chiamata al DOS per terminare un processo in esecuzione e ripresa del controllo da parte del processo padre.

**Funzione\_Chiusura equ 4Ch**

**Chiamata\_DOS equ 21h**

**Codice\_Di\_Ritorno DB ?**

Codice del programma

.....

.....

.....

**MOV AH, Funzione\_Chiusura**

**MOV AL, Codice\_Di\_Ritorno**

**INT Chiamata\_DOS**

La tabella seguente elenca le principali funzioni fornite dal DOS

AH	Descrizione	AH	Descrizione
01	Legge un carattere dal dispositivo standard di input (tastiera) STDIN	02	Scrive un carattere su dispositivo standard di output (video) STDOUT
05	Scrive un carattere sulla stampante	06	Input/Output diretto da console
07	Legge un carattere dal dispositivo standard di input (tastiera) STDIN senza eco	08	Legge un carattere dal dispositivo standard di input (tastiera) STDIN senza eco e gestisce il CTRL-BREAK
09	Scrive una stringa sul video STDOUT	0A	Legge una stringa da tastiera STDIN
0B	Restituisce lo stato del dispositivo standard di input STDIN	0C	Svuota il buffer del dispositivo standard di input STDIN
0D	Scarica buffer su disco	0E	Imposta il Drive di default
19	Fornisce il drive di default	25	Imposta il vettore di interrupt
2A	Fornisce la data del sistema	2B	Imposta la data del sistema
2C	Fornisce l'ora del sistema	2D	Imposta l'ora del sistema
2E	Imposta il flag di verifica scrittura su disco	30	Fornisce la versione del DOS in uso nel sistema
35	Legge il vettore di interrupt		
36	Fornisce lo spazio libero su disco	39	Crea una subdirectory (cartella)
3A	Cancella una subdirectory	3B	Imposta la directory corrente

AH	Descrizione	AH	Descrizione
3C	Crea un file	3D	Apri un file
3E	Chiude un file	3F	Legge caratteri da file
40	Scrive caratteri su file	41	Cancella file
42	Imposta il file pointer di un file	43	Imposta/Legge gli attributi di un file
47	Restituisce la directory corrente	4C	Termina l'esecuzione di un "programma"
4D	Legge il codice di ritorno dopo l'esecuzione di un processo	54	Legge il flag di verifica scrittura
56	Rinomina un file	57	Legge/Imposta la data del file

Tabella 1

---

AH = 01h – Legge un carattere dal dispositivo standard di INPUT, (tastiera), con eco

Azione

Legge un carattere dal dispositivo standard di input con eco. Se nessun carattere è disponibile aspetta fino a quando non ne viene immesso uno.

Ritorna AL = codice ASCII del carattere letto

Note

Gestisce il ^C/^Break. Se viene inviato il control-C oppure il Control break il processo viene interrotto

---

AH = 02h –Scrive un carattere sullo standard OUTPUT (video)

Azione

Invia al dispositivo standard di output un carattere.

In ingresso

DL deve contenere il carattere da visualizzare

Note

In AL viene ritornato l'ultimo carattere visualizzato cioè quello contenuto in DL.

Se DL=09, il codice di <TAB>, allora AL conterrà 32, il codice ASCII del carattere spazio (Space).

Il tabulatore viene tradotto in spazio.

Viene gestito il ^C/^Break

---

AH = 05h Scrive un carattere sulla stampante STDPRN

Azione

Invia un carattere al dispositivo di listing corrente (stampante LPT1)

In ingresso

DL contiene il codice del carattere da stampare

Note

La tastiera controlla se viene premuto ^C/^Break

STDPRN è la prima porta parallela (LPT1)

Se la stampante è occupata la funzione aspetta che si liberi.

Non vi è modo di controllare se la stampante è occupata.

---

AH = 06h – INPUT /OUTPUT diretto di un carattere senza bufferizzazione

Azione

Legge un carattere dal dispositivo standard di input oppure restituisce zero se nessun carattere viene digitato.

Scrive un carattere sul dispositivo standard di output.

In ingresso

DL contiene la funzione richiesta: da 0Ch a 0Fh=output che corrisponde al codice del carattere da visualizzare, se invece DL contiene 0Fh viene richiesto un input.

In AL viene ritornato il codice del carattere visualizzato

Ritorna

In output niente. In input azzera il flag di zero se il carattere è disponibile e AL contiene il codice del carattere digitato; se il dato non è pronto viene settato il flag di Zero (ZF).

Note

Non viene controllato il ^C/^Break

Se il codice del carattere ritornato è 00h, l'utente ha premuto un tasto funzione o di controllo ed il relativo codice viene fornito dalla successiva chiamata della funzione.

AH=07h – INPUT di un carattere senza eco

Azione

Legge un carattere dalla tastiera senza eco su video. Se nessun carattere è disponibile aspetta fino a quando non viene premuto uno.

Ritorna

in AL il carattere letto dal dispositivo di input standard

Note

Non gestisce il ^C/^Break

AH = 08h - INPUT di un carattere senza eco

Azione

Legge un carattere dalla tastiera senza copiarlo su video. Se nessun carattere è disponibile aspetta fino a quando ne viene premuto uno.

Ritorna

in AL il carattere letto dal dispositivo standard di input

Note

Gestisce il ^C/^Break.

AH = 09h – Scrive un stringa sullo standard OUTPUT (video)

Azione

Scrive una stringa su video

In ingresso

L'indirizzo iniziale della stringa da visualizzare deve essere contenuto in DS:DX e deve essere terminata dal carattere sentinella '\$' che non viene trasmesso.

Ritorna

AL= 24h che è il codice del carattere '\$.'

Note

Gestisce il ^C/^Break

AH = 0Ah – lettura di una stringa, in modo bufferizzato, dallo STDIN

Azione

Legge una stringa dal dispositivo standard di input corrente fino ad includere il carattere ASCII carriage return (0Dh). I caratteri ricevuti vengono memorizzati in un buffer definito dall'utente.

In ingresso

DS:DX devono puntare all'inizio dell'area di memoria utilizzata come buffer.

Note

Viene gestito ^C/^Break

Tipica sintassi di definizione del buffer:

NomeBuffer DB 30,?,30 DUP(?)

Nell'esempio viene definito un buffer di nome NomeBuffer la cui dimensione è 30. Si noti il ? cioè il byte successivo viene lasciato libero affinché venga immessa, dalla routine, la lunghezza corrente della stringa letta. I caratteri vengono inseriti a partire dall'indirizzo NomeBuffer +2

Formato del buffer

Offset rispetto all'indirizzo di partenza DS:DX	Dimensione	Descrizione
00	1	Dimensione del buffer
01	1	Numero di caratteri inserito escluso il carattere che termina l'immissione(CR)
02	N, max 255	Caratteri letti compreso il carattere <INVIO>

Tabella 2

---

AH=0Bh – Fornisce lo stato del dispositivo standard di Input STDIN

Azione

Controlla se un carattere è disponibile del dispositivo standard di input

Ritorna

AL = 00h se nessun carattere è disponibile

AL = FFh se il carattere è disponibile

Note

Viene gestito il ^C/^Break

AH = 0Ch – Pulisce il buffer di input e legge dal dispositivo di input

Azione

Pulisce il buffer del dispositivo standard di input e chiama le funzioni standard di input

In Ingresso

AL = numero della funzione da eseguire dopo aver pulito il buffer.

Le funzioni possono essere le 1,6,7,8,0Ah

Gli altri registri devono essere inizializzati in accordo alla funzione chiamata . Vedi sopra.

Ritorna

Se la funzione è 0Ah niente

Se la funzione è 1,6,7 in AL il codice del carattere letto

Note

Se AL non contiene uno dei seguenti valori 01h,06h,07h,08h, oppure 0Ah, il buffer viene pulito ma non viene chiamata alcuna funzione di input.

Lo scopo è quello di ignorare qualunque carattere digitato prima dell'esecuzione della chiamata.

---

AH = 0Dh – Reset del disco

Azione

Questa funzione scarica i buffer su disco: praticamente fa un aggiornamento del disco

Note

Se il programma non riesce a chiudere i file prima che il disco venga rimosso ed i file sono stati modificati le loro entry nella directory non sono corrette.

AH = 0Eh – seleziona il Drive di default

Azione

Imposta il drive di default (corrisponde al comando DOS `nomedrive: ;` esempio `c:` )

In ingresso

DL contiene il nuovo drive di default (0=A:, 1=B:, etc)

In AL viene restituito il numero di drive logici installati nel sistema.

AH = 19h – Fornisce il DRIVE di default o drive corrente

Azione

Fornisce il codice del drive corrente

Ritorna

in AL il codice del drive (0=A:, 1=B:, etc)

Nota

Alcune funzioni fanno partire la numerazione dei drive da 1 (per esempio la funzione 1CH che fornisce informazioni relative al drive corrente ed un puntatore al byte della FAT=File Allocation Table) riservando il valore 0 per il drive corrente.

AH = 25h - Set Interrupt Vector

Azione

Inizializza un vettore di interrupt a puntare ad un nuovo indirizzo

In ingresso

AL = numero di interrupt

DS:DX = indirizzo di partenza della routine (handler) che gestirà l'interrupt con codice (AL)

Note

prima di cambiare il contenuto del vettore di interrupt si dovrebbe usare la funzione 35H per ricavare l'indirizzo originale che dovrebbe essere reimpresso alla fine della routine di gestione (handler) dell'interrupt.

AH = 2Ah – Fornisce la data del sistema

Azione

Ritorna la data del sistema ed il giorno della settimana

Ritorna

CX = anno, che può andare dal 1980 al 2099, come differenza tra l'anno impostato e 1980; esempio se l'anno impostato nel sistema fosse 2002 in CX avremmo  $2002-1980=22$ ; DH = mese; DL = giorno; AL = giorno della settimana (00h=Domenica)

AH = 2Bh – Imposta la data del sistema

Azione

Reimposta la data dell'orologio del sistema

In ingresso

CX = anno data dalla differenza tra l'anno che si vuole impostare ed il 1980. L'anno che si impostare deve essere compreso nel range 1980-2099)

DH = mese DL = giorno

Ritorna

AL = 00 impostazione avvenuta con successo

AL=0FFh data non valida e la data del sistema non viene modificata

AH = 2Ch – Fornisce l'ora del sistema

Azione

Fornisce il tempo del giorno dell'orologio del sistema

Ritorna

CH = ora (0-23) CL = minuti (0-59) DH = secondi (0-59) DL = 1/100 di secondo (0-99)

AH = 2Dh – Imposta la data del sistema

Azione

Imposta la data del sistema

In ingresso

CH = ora CL = minuti DH = secondi DL = 1/100 di secondo

Ritorna

AL = 00h operazione eseguita con successo.

AL=0FFh ora non valida; aggiornamento non effettuato

AH = 2Eh – imposta il flag di verifica della scrittura su disco.

Azione

Imposta o cancella il flag di read after write: cioè se dopo una scrittura su disco si deve verificare se la scrittura sia avvenuta con successo o meno. La verifica viene fatta rileggendo quanto scritto.

In ingresso

AL = 0 il flag viene messo ad Off

AL =1 il flag viene messo ad On

Note

Questa funzione si preoccupa dell'integrità dei dati

Lo stato di default al boot è Off

AH=30h – Legge la versione DOS installata

Azione

Ritorna il numero della versione, del DOS, installata nel sistema

Ritorna

AL il numero più grande della versione (esempio 6.22:AL=6)

AH il numero minore della versione (esempio 6.22:AH=22)

AH=35h - Get Interrupt Vector

Azione

Ritorna il segmento:offset del vettore di interrupt associato al codice di interrupt fornito in ingresso

In ingresso

AL contiene il numero dell'interrupt

Ritorna

Viene restituito all'indirizzo ES:BX = indirizzo della routine di gestione dell'interrupt

Nota

Questo è il modo per leggere il contenuto del vettore di interrupt. Il contenuto originale del vettore di interrupt, dopo essere stato memorizzato può essere modificato chiamando la funzione 25h

AH = 36h – Fornisce il numero di cluster liberi in un disco (drive) specificato

In ingresso

DL=numero di codice del drive (0=default, 1=A:, etc)

Ritorna

Se il drive specificato è valido

AX = numero di settori per cluster BX = numero di cluster liberi CX = numero di byte per settore

DX = totale dei cluster del drive

Se il drive specificato non è valido

AX = FFFFh

Note

Lo spazio libero in byte è  $AX * BX * CX$   
 Totale numero di byte sul drive è  $AX * CX * DX$   
 I cluster rovinati sono considerati come utilizzati

AH = 39h - "MKDIR" – Crea una subdirectory (cartella)

Azione

Crea un subdirectory usando uno specificato pathname: comando MD del DOS.

In ingresso

DS:DX punta all'indirizzo della stringa che contiene il percorso ed il nome della cartella che si vuole creare. La stringa deve essere terminata con il carattere nullo (codice 0)

Ritorna

CF è azzerato se l'operazione è stata portata a termine con successo.

CF è settato se vi è stato un errore: il codice di errore viene messo in AX = codice di errore (03h,05h)

AX=3 se il percorso specificato non è valido oppure esiste già una cartella con lo stesso pathname.

AX=5 l'accesso è negato.

AH = 3Ah - "RMDIR" – Rimuove una cartella

Azione

Cancella una specificata cartella: comando RD del DOS

In ingresso

DS:DX puntano alla stringa, terminata con il carattere nullo, che rappresenta il pathname della directory che deve essere rimossa

Ritorna

CF azzerato se l'operazione è stata terminata con successo

CF settato se vi è stato un errore ed il codice di errore viene messo in AX.

Il codice di errore può essere:

3:percorso non trovato

5:accesso negato

6: si vuole eliminare la directory corrente

16: la directory non è vuota

AH = 3Bh - "CHDIR" – Imposta la directory corrente

Azione

Impone che una specificata subdirectory sia la subdirectory corrente: comando CD del DOS (CD=change directory)

In ingresso

DS:DX punta all'indirizzo della stringa che identifica il pathname della directory che dovrà diventare quella corrente. La stringa deve terminare con il carattere nullo

Ritorna

CF azzerato se l'operazione ha avuto successo CF settato in caso di errore con AX contenente il codice di errore che può essere 03h che vuol dire path non trovato

AH = 3Ch - "CREATE" – Crea o tronca un file

Azione

Crea un nuovo file, nella directory specificata del drive indicato, con il nome assegnato, o tronca la dimensione di un file esistente alla lunghezza zero.

Il file viene aperto per potervi accedere in lettura/scrittura. Viene restituito un descrittore, di 16 bit, del file (file handle)

handle=maniglia di accesso al file per potere effettuare le successive operazioni di lettura, scrittura.

Il file handle è univoco per ogni file aperto, creato o troncato.



In Ingresso

CX= attributi del file:

0:normale

1:a sola lettura

2:nascosto

3:di sistema

DS:DX punta all'indirizzo di memoria contenente il pathname del file da creare o troncato: il pathname deve essere una stringa terminata con il carattere nullo (codice=0)

Esempio: Patname DB "C:\Naso\MioFile",0

Ritorna

il flag CF azzerato se l'operazione è stata portata a termine con successo e in AX viene messo un file handle assegnato al file dal sistema per identificarlo univocamente nelle successive operazioni.

Praticamente è un nome interno che il sistema assegna al file

il flag CF settato se vi è stato un errore il cui codice viene impostato in AX. Esso può valere:

3:path non trovato

4: si è superato il limite dei file gestibile cioè si superato il numero messo nel comando FILES=n che si trova nel Config.sys

5:accesso negato

AH = 3Dh - "OPEN" – Apre un file esistente

Azione

Apre un file specificato in una directory del drive default o di un drive specificato.

Ritorna un handle a 16 bit per gli accessi futuri

In ingresso

AL = metodo di accesso

0:accesso in lettura

1:accesso in scrittura

2:accesso in lettura/scrittura

DS:DX punta alla stringa che identifica il pathname del file. Stringa che deve terminare con il carattere nullo

Ritorna

CF azzerato se l'operazione ha avuto successo ed AX contiene il file handle

CF settato in caso di errore con AX = codice di errore che può essere :,02h,03h,04h,05h,0Ch

2:file non trovato

3:path non esiste

4:troppi file aperti

5:accesso negato

0Ch:codice di accesso non valido

Note

Qualsiasi file di sistema o qualsiasi file nascosto che ha lo stesso nome del path indicato sarà aperto da questa funzione. Al ritorno il puntatore di lettura/scrittura del file verrà messo a zero. Cioè si parte dall'inizio del file

AH = 3Eh - "CLOSE" – Chiude file

Azione

Chiude un file aperto con successo. Tutti i buffer vengono scaricati su disco ed l'handle del file viene liberato per essere riutilizzato.

Se il file è stato modificato vengono aggiornate la data, l'ora e la lunghezza del file.

Ingressi

BX:file handle lo stesso fornito dal sistema al momento della creazione o della apertura del file

Ritorna

CF azzerato se l'operazione ha avuto successo

CF settato in caso di errore il cui codice viene messo in AX: codice di errore che può essere 06h in caso in cui il file non sia stato aperto.

AH = 3Fh - "READ" – Lettura di una sequenza di byte da un file o da un dispositivo di Input  
Azione

Legge uno specificato numero di byte da un file aperto con successo o da un dispositivo di input  
In ingresso

BX = file handle

CX = numero di byte che devono essere letti

DS:DX = l'indirizzo di partenza del buffer, in cui dovranno essere inseriti i caratteri letti, che dovrà avere una dimensione almeno pari al valore contenuto in CX

Restituisce

CF azzerato se l'operazione ha avuto successo - AX = numero di byte effettivamente letti (0 se viene trovato l' EOF prima della chiamata)

CF settato in caso di errore. AX = codice di errore (05h,06h)

05h: Accesso Negato

06h:handle non valido o file non aperto

Note

I dati vengono letti a partire dalla posizione corrente del file pointer (puntatore al byte corrente), il puntatore del file viene aggiornato dopo la lettura avvenuta con successo. Il valore contenuto in AX può essere minore di quello indicato in CX: nel caso di una lettura parziale in quanto si trova l'EOF  
Se si legge dalla CON (tastiera) la lettura termina dopo aver digitato il tasto <Invio>

AH=40h - "WRITE" – Scrive su un file o su un dispositivo di output

Azione

Scriva un numero di byte su un file aperto con successo o su un dispositivo

In ingresso

BX = file handle

CX = numero di byte da scrivere

DS:DX = indirizzo di partenza dell'area di memoria in cui sono contenuti i dati che devono essere scritti.

Ritorna

CF azzerato se la scrittura ha avuto successo. AX = numero di byte effettivamente scritti

CF settato in caso di errore .AX = codice di errore (05h,06h)

5:accesso negato

6:handle non valido o file non aperto

Note

Se CX è zero non viene scritto alcun byte , ed il file viene troncato.

I dati sono scritti a partire dalla posizione indicata dal file pointer che viene aggiornato dopo una scrittura con successo.

Se al ritorno AX < CX e CF=0 vuol dire che non tutti i caratteri sono stati scritti o che c'è stato un errore;per esempio il disco è pieno.

AH = 41H - "UNLINK" – Cancella (logicamente) un file

In ingresso

DS:DX = Indirizzo di partenza della stringa contenete il pathname del file da cancellare (il pathname deve essere terminato con il carattere nullo) non sono permessi caratteri jolly (\*,?,.)

Ritorna

CF azzerato se l'esecuzione è avvenuta con successo AL contiene il codice del drive che conteneva il file cancellato

CF settato in caso di errore. AX = codice di errore (02h,05h)

2:file non trovato o file a sola lettura

5:accesso negato

AH=42h - "SEEK" – imposta la posizione del file pointer o file position associato ad un file

Ingressi

AL = punto di partenza dal quale considerare lo spostamento 00h inizio file, 01h posizione corrente, 02h fine file : Se la posizione di partenza dello spostamento è l'inizio del file il valore deve essere positivo negli altri casi può essere negativo

BX = file handle

CX:DX = quanti byte si deve spostare il file pointer rispetto alla posizione di partenza

Ritorna

CF azzerato se lo spostamento ha avuto successo. DX:AX = nuovo valore, in byte, del puntatore al file dalla posizione di partenza del file

CF settato in caso di errore AX = codice di errore (01h,06h)

1:numero della funzione non valido

6:handle non valido o file non aperto

Note

Nel metodo 2 se si chiama la funzione con offset zero viene restituita la dimensione del file in byte  
Nei metodi 1 e 2 ci si può posizionare prima dell'inizio del file ma una successiva operazione di lettura o scrittura provocherà un errore.

AH=43 – Legge gli attributi del file

In ingresso

AL = 00h

DS:DX indirizzo di partenza della stringa contenente il pathname del file:la stringa deve terminare con il carattere nullo

Ritorna

CF azzerato se l'operazione ha avuto successo CX = contiene gli attributi (Vedi Tabella 3)

bit 5=archivio

bit 2=sistema

bit 1=nascosto(hidden)

bit 0= a sola lettura

CF settato in caso di errore, AX = codice di errore (01h,02h,03h,05h)

1:codice di funzione non valido

2:file non trovato

3:percorso o file non trovato

5:gli attributi non possono essere modificati.

AH=43 – Imposta gli attributi del file

In ingresso

AL = 01h

CX = nuovi attributi

bit 5=archivio

bit 2=sistema

bit 1=nascosto(hidden)

bit 0= a sola lettura

DS:DX indirizzo di partenza del pathname

Ritorna

CF azzerato se l'operazione è stata portata a termine con successo. AX viene distrutto

CF settato ; AX = codice di errore (01h,02h,03h,05h)

- 1:codice di funzione non valido
- 2:file non trovato
- 3:percorso o file non trovato
- 5:gli attributi non possono essere modificati

Campi di bit per gli attributi:

Bits	7	6	5	4	3	2	1	0
Descrizione	shareable	-	archive	directory	vol. label	system	hidden	read-only

Tabella 3

---

AH = 47h – Legge il nome della directory corrente

In ingresso

DL = numero del drive (00h = default, 01h = A:, etc)

DS:SI indirizzo di partenza in cui mettere il path della directory corrente . Buffer di destinazione (64 byte)

Ritorna

CF azzerato in caso di successo

CF settato in caso di errore. AX = Codice di errore (06h) Il drive specificato non è valido

Note

Il path restituito non contiene il drive ed il backslash iniziale (\) e viene terminato con il carattere nullo. Se la directory corrente fosse la radice verrebbe restituito come path il carattere nullo.

---

AH = 4Ch - "EXIT" – Termina il processo in corso (programma) e restituisce il controllo al COMMAND.COM o alla routine chiamante.

In ingresso

AL = restituisce il codice cioè si indica come è terminato il programma con successo o meno

Nota

Tutti i file aperti vengono chiusi e lo spazio di memoria assegnato viene liberato

Questo metodo è consigliato per terminare i programmi .exe

Il codice di ritorno può essere controllato attraverso la funzione 4Dh e anche attraverso il file di comandi (batch) attraverso la variabile d'ambiente ERRORLEVEL(IF ERRORLEVEL=codicedierrrore).

Convenzionalmente il codice di ritorno zero indica il caso di funzione terminata con successo qualunque altro valore se termina perché è intervenuto un errore durante l'esecuzione della funzione.

I codici standard del DOS sono:

0: operazione terminato con successo

1: terminata con CTRL-BREAK

2:errore critico

3:TSR (Terminate and stay resident): Programmi residenti.

I codici di ritorno sono a discrezione del programmatore (evitare i valore da 1 a 3). Se si vuole utilizzare l'istruzione "IF ERROR LEVEL " utilizzare il registro AL per restituire il codice.

---

AH = 4Dh – Legge il codice di ritorno dall'esecuzione di un programma (ERRORLEVEL)

Ritorna

AH =Contiene il modo in cui un programma è stato terminato (00=normal, 01h control-C abort, 02h=critical error abort, 03h terminate and stay resident) AL = Codice di ritorno

Note

La parola in cui il DOS memorizza il codice di ritorno viene azzerata dopo la chiamata così che il codice di ritorno può essere letto solo una volta.

COMMAND.COM memorizza il codice di ritorno dell'ultimo comando esterno eseguito nella variabile d'ambiente ERRORLEVEL

AH = 54h – Legge lo stato corrente del flag di verifica scrittura (VERIFY)

Ritorna

AL = verify flag (00h=off, 01h=on, cioè tutte le scritture su disco vengono verificate dopo lo scrittura)

AH = 56h - "RENAME" – Rinomina file

In ingresso

DS:DX punta al pathname del file che deve essere rinominato (il path deve finire con il carattere nullo) non sono permessi i caratteri jolly

ES:DI punta al pathname che identifica il nome nuovo

CL = maschera degli attributi

Ritorna

CF azzerato se l'operazione ha avuto successo

CF settato in caso di errore , AX= codice di errore (02h,03h,05h,11h)

2:file non trovato

3:path non trovato o il file non esiste

5:accesso negato

11: Il nuovo nome ha un path che porta su un dispositivo diverso

Note

Permette lo spostamento in directory della stessa unità logica.

Non imposta l'attributo di archivio

I file aperti non si possono rinominare

Si possono rinominare directory

AH = 57h – Legge la data e l'ora dell'ultima modifica del file

In ingresso

AL = 00h BX = file handle

Ritorna

CF azzerato se l'operazione ha avuto successo: CX = contiene l'ora: DX = contiene la data

CF settato se vi stato un errore , AX = codice di errore (01h,06h)

1:codice della funzione non valido

6:file handle non valido

Bit di campo per l'ora:

Bit	15-11	10-5	4-0
Descrizione	ora	minuto	secondo

Tabella 4

Bit di campo per la data

Bit	15-9	8-5	4-0
Descrizione	Anno relativo al 1980 cioè se l'anno è 2002 troveremo 2002-1980=22	mese	giorno

Tabella 5

AH = 57h – Imposta la data e l'ora dell'ultima modifica

In ingresso

AL = 01h

BX = file handle

CX = La nuova ora : i bit di campo sono gli stessi della chiamata precedente

DX = La nuova data: I bit di campo sono gli stessi della chiamata precedente

Ritorna

CF azzerato se l'operazione ha avuto successo

CF settato in caso di errore AX = codice di errore (01h,06h)

1: codice della funzione non valido

6: file handle non valido

<b>Titolo:</b>	Interrupt 21h
<b>Autore:</b>	Michele Naso
<b>Email:</b>	michele.naso@istruzione.it
<b>Classe:</b>	QUINTA INFORMATICA (5IA)
<b>Anno scolastico:</b>	2003/2004
<b>Scuola:</b>	Itis Euganeo Via Borgofuro, 6 Via Borgofuro 6 - 35042 Este (PD) - Italy Telefono 0429.21.16 - 0429.34.72 Fax 0429.41.86 <a href="http://www.itiseuganeo.it">http://www.itiseuganeo.it</a> <a href="mailto:informazioni@itiseuganeo.it">informazioni@itiseuganeo.it</a>
<b>Note legali:</b>	Diffusione consentita con obbligo di citare le fonti