



## i trucchi del CP/M

a cura di Pierluigi Panunzi

### I messaggi d'errore (3): DDT - SUBMIT - ASM

*Siamo giunti alla terza puntata della serie riguardante i messaggi d'errore generati dai programmi che girano sotto CP/M: in questo numero vedremo i messaggi diagnostici generati da tre programmi molto ben conosciuti in quanto presenti nella quasi totalità dei sistemi che girano sotto CP/M e cioè il DDT (Dynamic Debugging Tool), il SUBMIT e l'ASM.*

#### I messaggi d'errore del DDT

A dispetto della grande potenza e ricchezza di comandi di questo programma, la sua diagnostica si riduce ad appena due messaggi d'errore: «?» e «??=».

In particolare il primo «carattere d'errore» viene emesso in tre occasioni differenti, per cui a seconda del contesto in cui ci si trova si avrà una conseguente spiegazione per l'errore commesso e sui provvedimenti da prendere.

La prima delle tre occasioni in cui appare il «?» è quella in cui il DDT non riesce a trovare nella directory il file che desideriamo caricare in memoria, con lo scopo di effettuarne l'analisi.

In questo caso il provvedimento più ovvio da prendere è quello di controllare se nella digitazione del nome del file non abbiamo commesso un errore, mentre in seconda battuta conviene uscire dal DDT e controllare se effettivamente il file desiderato esiste o meno: attenzione che se il file ha l'attributo «System», allora non può essere visibile con il comando DIR, mentre viceversa lo sarà usando opportunamente lo STAT.

La seconda occasione in cui può essere inviato un «?» sullo schermo, in risposta ad un certo nostro comando, riguarda appunto un comando che si

riferisce ad un file di tipo «.HEX»: in questo caso il «?» indica che il DDT ha riscontrato un errore all'interno del file in questione, ad esempio un'errata «check-sum», oppure un campo errato in certo record.

In questo caso, come si era già visto parlando del programma PIP, conviene «buttare a mare» il file incriminato a rigenerarlo a partire dal programma sorgente in Assembler: se però quest'ultimo non è più disponibile allora sarebbe consigliabile dedicarsi ad altre attività, senz'altro più proficue...

La terza occasione in cui il DDT mostra il «?» è quella in cui, dopo aver impostato il comando «A» (Assemble), impostiamo ad un certo punto un'istruzione non riconosciuta dal DDT e perciò dall'Assembler 8080: bisogna controllare perciò innanzitutto la corretta impostazione dell'istruzione, ricordando che il DDT in questa occasione non accetta i «TAB», né numeri esadecimali seguiti dal carattere «H», che viceversa deve essere omissso, dato che la «radice numerica» utilizzata dall'Assembler del DDT è quella esadecimale.

L'altro tipo di messaggio d'errore («??=») viene emesso dal DDT quando si è impostato il comando «L» (List disassembled), per analizzare appunto il disassemblato delle istruzioni contenute in una certa zona di memoria.

Dal momento che l'assembler dell'8080 non utilizza come codici di istruzioni («opcode») tutti i 256 byte possibili con 8 bit, ecco che, incontrando un byte non corrispondente ad alcuna istruzione dell'8080, il DDT visualizza «??=» seguito dal valore esadecimale che ha generato l'errore.

Un esempio molto frequente di tale possibilità si ha quando si disassembla un programma scritto per lo Z80, che, come noto, sfrutta per le sue nuove istruzioni i «buchi» lasciati vuoti dall'8080; altra possibilità è quella di essere capitati per errore in una zona di

memoria contenente delle stringhe di caratteri ASCII, per cui già dall'analisi del «listing» risultante si può avere il sentore che ci sia qualcosa che non va, specie se si conosce già bene l'Assembler.

#### I messaggi d'errore di SUBMIT

Nel caso di questo programma abbiamo 9 tipi differenti di messaggi d'errore e prima di analizzarli in dettaglio ricordiamo che il programma SUBMIT consente di eseguire automaticamente una sequenza di comandi contenuti in un certo file, ottenendo lo stesso funzionamento che otterremmo impostando uno dopo l'altro, i singoli comandi dalla tastiera, senza doversi perciò preoccupare dell'esattezza della sequenza di comandi nonché della loro corretta digitazione.

Analizziamo dunque i singoli messaggi d'errore.

##### Messaggi d'errore di SUBMIT

```
CANNOT CLOSE, READ ONLY?
COMMAND BUFFER OVERFLOW
COMMAND TOO LONG
DIRECTORY FULL
DISK WRITE ERROR
ERROR IN LINE n
INVALID CONTROL CHARACTER
NO «SUB» FILE PRESENT
PARAMETER ERROR
```

#### CANNOT CLOSE, READ ONLY?

Il programma SUBMIT emette sul video questo primo messaggio d'errore nel caso in cui risulti protetta fisicamente da scrittura l'unità a dischi sulla quale sta tentando di scrivere il file di output, il ben noto «SSS.SUB».

Può capitare infatti che il disco dove si trovi il file contenente i comandi sia protetto dalla scrittura, perché ad esempio di «sistema», ed in tal caso il SUBMIT non può creare il file di appoggio: ecco che conviene copiare sull'unità B: il file di comandi ed invoca-

re il SUBMIT indicando che il file in oggetto è appunto su B:

Su tale unità verrà dunque generato il file d'appoggio \$\$\$SUB.

### COMMAND BUFFER OVERFLOW

Questo messaggio d'errore appare quando il file di comandi è troppo lungo per i 2K byte di buffer disponibili all'interno del SUBMIT.

In questo caso si può rimediare all'inconveniente spezzando in due o più parti il file di comandi, ponendo come ultima linea del primo il «SUBMIT» del secondo file e così via, ottenendo così una concatenazione o meglio un «nesting» di comandi.

### COMMAND TOO LONG

Analogo al caso precedente è questo, in cui stavolta non è tanto il file troppo lungo, ma un singolo comando: bisogna infatti ricordarsi che la lunghezza massima del singolo comando è di 125 caratteri, mentre in caso contrario il SUBMIT non eseguirà il comando stesso, ma si fermerà indicando appunto l'accaduto, senza però specificare quale era la linea di comando troppo lunga.

Bisogna a questo punto rileggersi il file di comando per trovare la riga incriminata per cercare di accorciarla ad esempio rinominando i file in essa citati con nomi più corti ed usando abbreviazioni laddove ciò sia consentito.

### DIRECTORY FULL

Tanto il messaggio quanto le azioni da intraprendere sono così ovvie che non ci soffermiamo oltre se non per ricordare che il disco può anche non essere pieno, ma si può essere rovinata la directory, per cui il disco può apparire pieno mentre in realtà non lo è.

### DISK WRITE ERROR

Anche questo messaggio è molto chiaro e sarà in generale preceduto da un analogo messaggio del BIOS: non ci soffermeremo oltre.

### ERROR ON LINE n

In questo caso il SUBMIT segnala al non accorto operatore che nella linea «n» del file di comandi c'è un comando che non è in grado di eseguire, perché magari il comando stesso contiene dei caratteri di controllo invisibili e cioè non ASCII: anche in questo caso si deve ri-editare il file ed in particolare la linea «n» stando più attenti. Ricordiamo che alla prima linea del file di comando è associato il numero di linea 001.

### INVALID CONTROL CHARACTER

Anche per questo messaggio ci troviamo in una situazione analoga alla

precedente, nella quale il SUBMIT incontra in una linea di comando un carattere di controllo, del tipo ^x (dove «x» è una qualsiasi lettera), non riconosciuto e perciò non eseguibile.

### NO «SUB» FILE PRESENT

È questo il messaggio d'errore che il SUBMIT mostra quando non trova nella directory corrente il file di comandi specificato nella linea di comando stessa, subito dopo l'invocazione del SUBMIT stesso.

### PARAMETER ERROR

L'ultimo messaggio d'errore viene mostrato quando il SUBMIT trova un carattere «\$» non seguito da un numero: ricordiamo infatti che i parametri che si possono passare al file di comandi sono in quest'ultimo identificati con «\$n» dove appunto «n» è una cifra compresa tra 0 e 9.

Ricordiamo invece che se vogliamo veramente impostare il carattere «\$» allora dovremo impostarlo come «\$\$».

### I messaggi d'errore di ASM

Questo programma, che come ben noto serve per assemblare un programma scritto in linguaggio assembly per il microprocessore 8080, possiede 7 tipi di messaggi d'errore, che ora andiamo ad analizzare.

#### Messaggi d'errore di ASM

CANNOT CLOSE FILES  
NO DIRECTORY SPACE  
NO SOURCE FILE PRESENT  
OUTPUT FILE WRITE ERROR  
SOURCE FILE NAME ERROR  
SOURCE FILE READ ERROR  
SYMBOL TABLE OVERFLOW

### CANNOT CLOSE FILES

Questo messaggio d'errore viene visualizzato allorché l'ASM non riesce a «chiudere» i suoi file di output o perché il dischetto è protetto fisicamente dalla scrittura o se è sopraggiunto un problema hardware che impedisce di completare la funzione di scrittura e/o di chiusura di uno dei file di output.

### NO DIRECTORY SPACE

Questo tipo di messaggio, identico ad uno di quelli del programma PIP, è ancora una volta un messaggio ben chiaro, per cui non aggiungiamo altro alla sua traduzione letterale «non c'è più spazio nella directory».

### NO SOURCE FILE PRESENT

Tale messaggio d'errore viene emesso dall'ASM quando quest'ultimo non trova nella directory di default il file

sorgente indicato nella linea di comando.

Bisogna ricordare che il programma ASM usa il nome del file impostato nel comando e successivamente ricerca su dischetto il file avente quel nome, ma con l'estensione «.ASM».

Invece nella linea di comando bisogna stare attenti a non porre «ASM» come estensione del file sorgente, in quanto i tre caratteri che si pongono dopo il «.» del nome del file rappresentano rispettivamente le unità logiche a dischi dove è posto il file sorgente e dove verranno posti successivamente i file di tipo «.HEX» e di tipo «list» generati dall'assemblatore.

### OUTPUT FILE WRITE ERROR

Come in altri casi analoghi, il messaggio in esame viene visualizzato allorché il BDOS riscontra un errore da parte della routine di «write» su disco: però nel caso in cui il BIOS possiede una logica di recovery degli errori da disco, tale messaggio non verrà mai visualizzato.

### SOURCE FILE NAME ERROR

Il messaggio in esame viene visualizzato quando nella linea di comando di attivazione dell'ASM all'interno del nome del file vengono utilizzati i caratteri «\*» o «?», tipici dei nomi di file ambigui, non accettati dall'Assembler, dal momento che questo deve trovare un unico file da processare.

### SOURCE FILE READ ERROR

Anche in questo caso si tratta di un messaggio molto chiaro, che si riferisce al caso in cui l'ASM ha dei problemi nel leggere il file sorgente, ad esempio per la presenza di qualche carattere di controllo «strano».

In questo caso conviene usare il comando TYPE come pure un text editor per vedere sul video il file sorgente e trovare così il punto che ha generato il blocco da parte dell'ASM.

### SYMBOL TABLE OVERFLOW

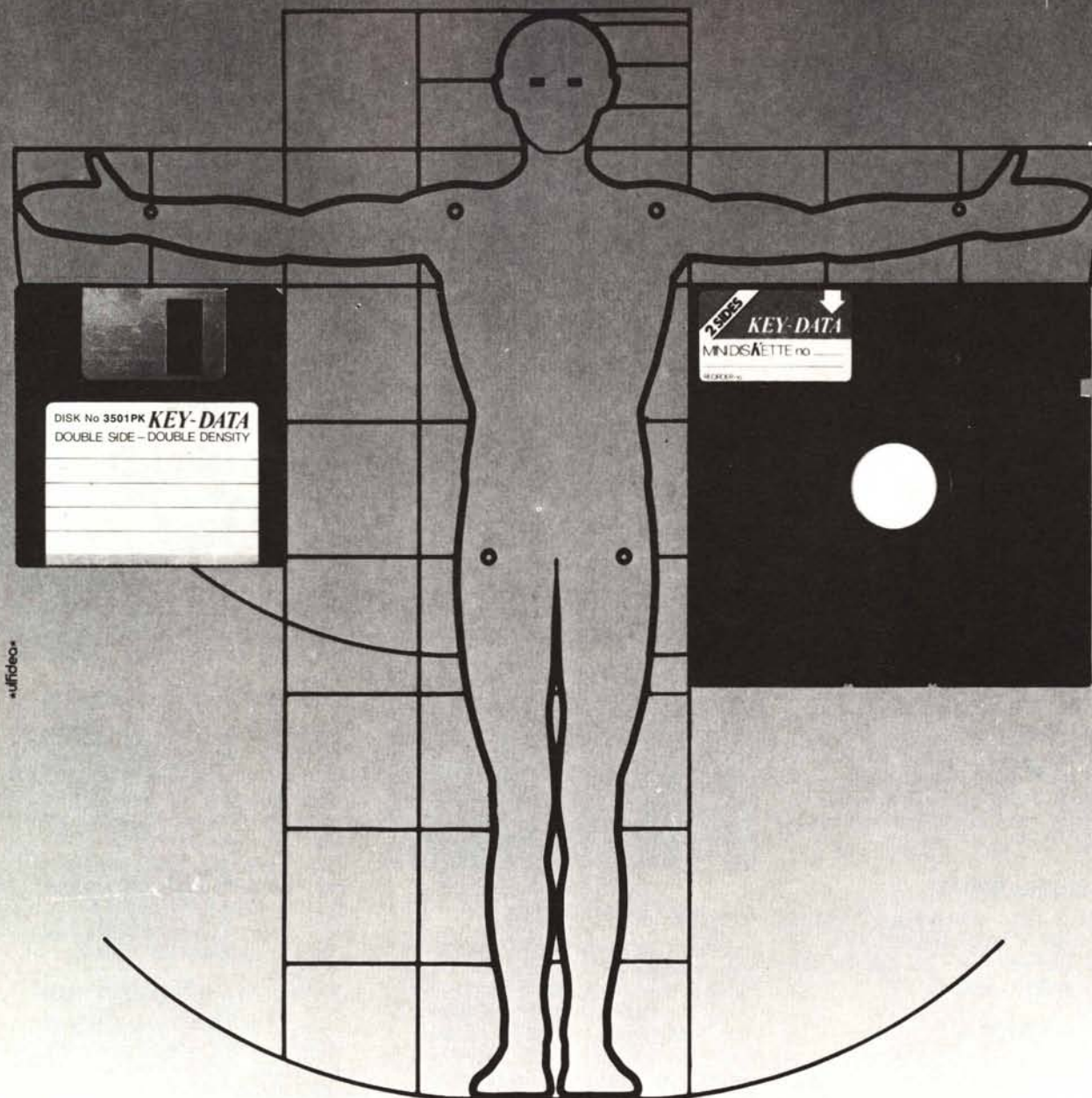
Ecco finalmente (è l'ultimo...) un messaggio non dovuto a particolari condizioni d'errore da parte dell'hardware del sistema o ad errate impostazioni dell'operatore: si tratta infatti di un messaggio dovuto ad un numero eccessivo di simboli (nomi di variabili, etichette, ecc.) usati nel file sorgente.

Per rimediare a tale errore bisogna spezzare il file sorgente in più parti, tutte da collegare con un sapiente ed opportuno uso della direttiva «ORG», in modo tale da riottenere un file oggetto correttamente allocato in memoria: il file sorgente riordinato in tale modo deve poi ovviamente essere ri-assemblato.



# KEY-DATA

L'EVOLUZIONE CHE GIRA NEL TEMPO



**GARANZIA**  
TOTALE 5 ANNI CON  
LA TECNOLOGIA DEL  
FUTURO PER LA  
PRODUZIONE  
ESCLUSIVA DEL DISCO  
KEY DATA.

**DURATA**  
DI OLTRE 50 MILIONI DI  
PASSAGGI CON TRACCE  
ERROR-FREE AL 100%.

**SICUREZZA**  
"OLD SYSTEM" CHE  
GARANTISCE LUNGA  
DURATA ALLA TESTINA  
DEL FLOPPY E AI DATI  
PROGRAMMATI DI  
RIMANERE NEL TEMPO.

**gi erre** informatica

DIVISIONE ACCESSORI E PERIFERICHE

GI-ERRE INFORMATICA s.r.l.  
42100 REGGIO EMILIA VIA UMBRIA 36/A  
TEL. 0522/38655-512345-512396-512301 TLX 530198 PP RE I  
70125 BARI VIA MONTE S. MICHELE 2/B TEL. 080/682107  
95100 CATANIA P.ZZA GALATEA 2 TEL. 095/375222