



i trucchi del CP/M

a cura di Claudio Rosazza

Ancora sul \$\$\$ SUB

Nella maggioranza dei casi i programmi applicativi vengono scritti con linguaggi di tipo evoluto, siano essi interpreti o compilatori.

In tal modo l'utente riesce a comporre procedure logiche molto complesse in tempi relativamente brevi e con minimo dispendio di energia. Considerato, però, che la flessibilità operativa spesso si paga con una decadenza delle prestazioni in termini di velocità, si comprende il perché certe applicazioni vengono realizzate direttamente in linguaggio macchina.

È il caso, per esempio, dei programmi di sort dove il requisito fondamentale è essenzialmente la velocità.

Usando un calcolatore con sistema operativo CP/M il mercato software mette a disposizione dell'utente vari tipi di programmi di ordinamento di file su disco che hanno più o meno tutti lo stesso difetto: l'interfacciamento software con il linguaggio evoluto. Fanno eccezione quei programmi di sort realizzati espressamente per quel particolare tipo di linguaggio (vedi l'MSORT Microsoft per il compilatore COBOL).

Prendiamo ad esempio il SUPER-SORT della Micropro.

Probabilmente è uno dei migliori programmi di questo tipo attualmente disponibili e viene contemporaneamente fornito in una versione direttamente eseguibile che richiede la digitazione dei parametri di ordinamento da tastiera ed in una versione

sotto forma di subroutine che dovrebbe consentire il collegamento con linguaggi evoluti ed il passaggio diretto dei parametri fra il programma applicativo e il programma di ordinamento.

In effetti il collegamento del Super-Sort sotto forma di subroutine con alcuni linguaggi evoluti, soprattutto compilatori, è possibile anche se non è decisamente semplice capirne il funzionamento al primo approccio. Esistono, però, casi in cui l'interfacciamento software richiede un lavoro estremamente pesante che in alcuni casi non assicura neanche una discreta affidabilità; è il caso degli interpreti che hanno quasi sempre grosse difficoltà ad interfacciarsi con programmi assembler. L'uso invece del Super-Sort sotto forma di programma eseguibile è estremamente semplice e richiede la digitazione dei parametri coinvolti nell'ordinamento di un file quali il nome del file da ordinare, il nome del file ordinato, la/le chiavi di ordinamento etc. Come fare per lanciare il programma di sort senza digitare i parametri in quanto calcolati e passati da un programma scritto in interprete Basic?

Ricorriamo ancora una volta al \$\$\$ SUB generandolo in ambiente Basic. La subroutine descritta in questa pagina non è altro che il programma SUBMIT.COM scritto in Basic.

Ammettiamo che dall'interno di un programma Basic desideriamo uscire per effettuare un sort di un file e quindi rientrare nel Basic eseguendo il menu principale della procedura.

Dovremmo creare un \$\$\$ SUB che sia la

codifica dei seguenti comandi CP/M:
SORT <lista di parametri >
MBASIC MENU

Per generare tale \$\$\$ SUB tramite la subroutine riportata è sufficiente comporre una stringa per ogni riga di comando chiamandola Z2\$ e passarla alla subroutine entrando in 15000; dopo aver passato l'ultima stringa occorre passare una Z2\$ di valore "F" per forzare la chiusura del file \$\$\$ SUB.


Occorre, però, tenere presente che i comandi vanno passati nell'ordine inverso a come dovranno poi essere eseguiti da parte del CP/M.

Il programma per generare i comandi di cui sopra sarà pertanto il seguente:

```
10 Z2$="MBASIC MENU": GOSUB 15000
20 Z2$="SORT <lista di parametri > ": GOSUB 15000
30 Z2$="F": GOSUB 15000
40 SYSTEM
```

L'istruzione SYSTEM consente di abbandonare il Basic tornando al CP/M che partirà ad eseguire i comandi contenuti nel \$\$\$ SUB appena creato. Nel caso che su quel disco si sia implementata la procedura di autoload descritta il mese scorso, la prima stringa da passare, che poi sarebbe l'ultimo comando da eseguire, dovrà essere sostituita con SUBMIT X

in modo da rigenerare l'autoload che sfrutta anch'esso il \$\$\$ SUB. Le stringhe di comando non possono essere più lunghe di 123 caratteri ed in particolare l'ultima stringa, cioè il primo comando, non più di 122 altrimenti la subroutine abortisce in uno STOP.

La subroutine è scritta in MBASIC Microsoft, ma può essere facilmente riscritta per qualsiasi linguaggio sia esso interprete o compilato o può essere usata in tutti quei casi in cui si desideri richiamare uno o più programmi .COM in sequenza dall'interno di un programma applicativo per poi tornare al programma applicativo stesso. L'unica limitazione è data dal fatto che il programma applicativo può passare dei parametri al programma .COM tramite la stringa di comando, ma il programma .COM non può viceversa passare parametri al programma applicativo. 

```
15000 IF Z5$="I" OR Z5$="E" THEN 15004
15001 Z5$="I"
15002 ZZ=123
15003 OPEN "0",#3,"$$$ SUB"
15004 IF LEN(Z2$)>ZZ THEN STOP
15005 IF Z2$="F" THEN 15010
15006 Z3$=CHR$(LEN(Z2$))
15007 Z4$=Z3$+Z2$+CHR$(0)+" "+5SPACE$(ZZ-LEN(Z2$))
15008 PRINT #3,Z4$
15009 IF Z5$="E" THEN 15011 ELSE RETURN
15010 Z5$="E":ZZ=122:RETURN
15011 CLOSE 3
15012 RETURN
```

**E' nato il
PERSONAL COMPUTER
che cambia
il modo di vedere
i personal.**


PRONTO, CELDIS?! TRATTO DA ANNI I PERSONAL, MA ORA TUTTI
VOGLIONO IL PERSONAL DELLA DIGITAL! COME POSSO FARE PER
DIVENTARE SUBITO RIVENDITORE AUTORIZZATO DIGITAL?

Semplice.



**Telefonaci:
(02) 6120041**

CELDIS

Distributore  Autorizzato
PERSONAL COMPUTERS*



* Solo per rivenditori sul territorio nazionale

Celdis Italiana S.p.A.

Via F.lli Gracchi, 36 - 20092 Cinisello B. (Mi) - Tel. (02) 612.00.41

Filiali:

10136 Torino - Via Mombarcaro, 96 - Tel. (011) 35.93.12/35.93.69

35100 Padova - Via Savelli, 15 - Tel. (049) 77.20.99/77.21.35

40138 Bologna - Via Massarenti, 219/4 - Tel. (051) 53.33.36

00162 Roma - Via G. Pitre, 11 int. 1 - Tel. (06) 42.38.55/427.15.50