

& DEBUG

di Stefano Laporta - Bologna

I lettori più attenti avranno notato una certa familiarità nel nome dell'Autore, già apparso parecchie volte nella rubrica Software SOA di Pierluigi Panunzi. Grazie all'acquisto di un Lemon II ha potuto sviluppare, nel tempo eccezionalmente breve di tre mesi, una notevole mole di programmi in assembler che ci ha immediatamente inviato. Tra questi abbiamo scelto, per la prima pubblicazione, un programma di Debug che permette di ottenere, oltre all'ormai famigerata Cross Reference, anche due liste delle variabili usate dal programma, una in ordine alfabetico ed una in ordine di "memorizzazione".

Cosa fa

Il programma di debug, che si usa con l'istruzione & seguita da un numero, permette le seguenti opzioni:

&1 — esegue la CROSS REFERENCE MAP, scrive cioè i numeri delle variabili usate dal programma e il numero di riga in

cui compaiono. Se una variabile appare più volte su una stessa riga viene indicato quante volte compare (es. K1 03*250 significa che la variabile K1 compare tre volte nella riga 250).

Vengono anche indicate le Function precedute da FN e le variabili Dummy (false) contenute nella Function che risultano precedute dalla AT (@). Una certa lentezza di esecuzione (qualche secondo) è dovuta al fatto che, per risparmiare spazio in memoria, viene scandito il programma Basic per ciascuna variabile.

L'ordine di presentazione è: prima le variabili semplici, poi i vettori quindi le stringhe e le Function; sempre in ordine alfabetico.

& — esegue la Variables List ovvero la lista delle variabili nello stesso ordine della cross reference, ma invece del numero di riga viene mostrato il valore attuale o la scritta "UNUSED" per quelle non usate. Per i vettori e le matrici viene stampata la dimensione e un numero di elementi a scelta. Scelta che si esegue facendo seguire il comando & dalla lettera A e dal numero di

elementi desiderati (es. &A10 stampa i primi 11 elementi, compreso quello di indice 0). Il valore di default per i vettori è tre che corrisponde ai primi quattro elementi.

Naturalmente le matrici vengono stampate come vettori così come sono in memoria; questo non è uno svantaggio, anzi permette di controllare il funzionamento corretto di istruzioni tipo la RECALL o di programmi che operano prodotti e trasposte sulle matrici.

&3 — viene stampata la Variables Table che si distingue dalla precedente solo per l'ordine delle variabili; queste infatti si presentano nello stesso ordine in cui sono state create dal programma. Dal momento che ogni volta che l'interprete deve usare una variabile scandisce sequenzialmente tutta la tabella finché non la trova, è chiaro che, spostando in alto le variabili più usate si ottiene un notevole incremento della velocità di esecuzione. Per spostare le variabili basta mettere una riga in testa al programma in cui si scrivono tutte le variabili da "velocizzare" in ordine: per esempio

10 A = X = I = K1 = S = M0 ecc.

&2 — esegue la cross reference più la Variables List e stampa anche quante variabili sono presenti nel programma.

Tutti questi comandi possono essere usati durante l'esecuzione semplicemente scrivendoli in una riga del programma, in tal caso viene stampato in alto a sinistra il

J&1 #65378	J&A5 #65378	J&3 #65378
----- CROSS REFERENCE MAP	----- VARIABLES LIST	----- VARIABLES TABLE
<pre>A 430 440 1010 1020 1050 1210 1220 AX 900 1040 04*1070 1090 BN 1020 1030 E 415 1070 1095 1210 EM 105 250 02*300 490 500 870 EX 460 03*490 495 860 F1 03*490 F2 03*1250 F3 03*1330 H 1000 02*1010</pre>	<pre>A =.0966801624 AX =30 BN =.0118632285 E =0 EM =5 EX =3 F1 =1 F2 [UNUSED] F3 [UNUSED] H =100 I =59 J =4 K =1 M =300</pre>	<pre>A\$=***** K=1 PG=3.1415 EM=5 NP=100 UM=5 MA=5 MS=300 SS=100 U1=.4 U2=.6 TM=43021 EX=3 I=59 AX=30</pre>
a	b	c

Esempio di output (parziale) del programma & DEBUG: cross reference (a) e lista delle variabili in ordine alfabetico (b) e di memorizzazione (c).

Il minifloppy con il programma &DEBUG per Apple II (codice DA2/02) può essere acquistato presso la redazione al prezzo di lire 15.000 (compresa IVA e spedizione).

Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) alla Technimedia srl, Via Valsolda 135, 00141 Roma.

Trenta sono troppi

Nell'Apple si passa da un eccesso all'altro: il Basic riconosce per i nomi delle variabili solo i primi due caratteri, spesso un po' pochini, il DOS invece accetta (anzi quasi pretendendo) trenta caratteri tutti significativi.

È decisamente comodo poter scrivere direttamente nel nome del file un gran numero di informazioni, come la data di creazione o gli indirizzi di memoria dei file binari, ma d'altra parte è decisamente scomodo dover riscrivere tutta questa roba ogni volta che si vuole ricaricare il file dal disco.

Anche il metodo di portarcisi sopra con il cursore e di ricopiare la riga non è proprio il massimo della praticità.

Molto più comodo sarebbe poter ridurre il numero di caratteri significativi pur lasciando la possibilità di scrivere sempre fino a trenta caratteri.

Il numero di caratteri significativi all'individuazione di un certo file si trova ovviamente dentro al DOS, e più precisamente dentro al segmento di DOS che esegue la OPEN del file. L'apertura di un file significa la localizzazione sul disco (o la allocazione nel caso di un nuovo nome) di tutte le informazioni necessarie al DOS per accedere, sia in scrittura che in lettura, al file in questione. In pratica, qualsiasi operazione di scrittura/lettura sul disco per prima cosa deve aprire il file su cui accedere. La routine che apre un file si chiama OPEN, e preleva il nome del file dal buffer Primary Name (locazioni \$AA75, \$AA92) e byte

dopo byte lo confronta con i nomi dei file prelevati via via dalla Directory. Se non trova nessun nome uguale (caso di un nuovo file) cerca sul disco uno spazio libero e prepara una nuova Directory che contenga il nome nuovo e abbia tutti i puntatori necessari per localizzare il file. Se invece uno dei nomi corrisponde a quello del Primary Name Buffer legge dalla Directory le informazioni necessarie a trovare il file in questione.

Il confronto byte per byte dei due nomi, quello letto sul CATALOG e quello battuto da tastiera che si trova nel buffer, ha termine solo in due casi: o un carattere è differente (nomi diversi) oppure sono stati confrontati, con successo, trenta caratteri. La locazione \$B203 (dec. 45571) contiene appunto il numero di caratteri significativi.

Se quindi provate a fare una POKE 45571,8 scoprirete che, pur potendo scrivere nomi lunghi trenta caratteri, solo i primi otto servono a distinguere il file. Nel caso che due file abbiano i primi otto caratteri uguali verrà sempre trovato solamente il primo in ordine di Catalog. La cosa può essere pericolosa e otto caratteri sono sempre tanti! Allora perché non rendere significativi solo i caratteri battuti da tastiera?

Per fare ciò si deve modificare la routine che scandisce i comandi DOS alla ricerca delle parole chiave, dei parametri tipo D2,S1 ecc. e appunto dei nomi di file. Questa routine, che si trova a \$A193, cerca nella riga le virgole e il CR (RETURN); una volta trovato ritorna al chiamante.

Ebbene ci basta fare in modo che una volta incontrato il RETURN, prima di rientrare, depositi nella locazione \$B203 il numero di caratteri del nome che sono stati battuti. Questo numero si trova nel registro Y, in quanto è servito per ricopiare i caratteri del nome dal buffer di tastiera \$200 al buffer di Primary Name. In questo modo si rendono significativi solo il numero di caratteri equivalente a quelli battuti da tastiera.

Questo ci permette di caricare un file anche solo con la prima lettera del nome, a patto che non ci siano altri file con la stessa iniziale che lo precedono sul catalog. Se ci sono si dovranno usare due lettere o tre o quattro; insomma tante quante ne servono a distinguere inequivocabilmente il file da quelli che lo precedono. Nel caso in cui un nome completo corrisponda ad una parte di uno che lo precede, ad esempio vogliamo il file PROVA ma sul CATALOG compare prima il file PROVA NUMERO UNO, basta battere tra PROVA e RETURN un paio di spazi per distinguerlo dall'altro.

Si possono usare le abbreviazioni per tutti i comandi che usano un nome (per esempio LOCK, DELETE, BRUN, VERIFY, OPEN ecc.).

Un solo comando è stato "rovinato" da questa modifica: la RENAME. Infatti quando noi premiamo il RETURN il numero di caratteri significativi è quello relativo alla somma dei due nomi e la open non riesce a trovare nessun file con queste caratteristiche. A questo punto vista la non grave perdita (la RENAME non è certo il comando più usato) e visto oltretutto che ci serviva dello spazio dentro al DOS per depositarvi la routine di input modificata, si è deciso di disabilitare la RENAME (sostituendola con una innoqua CLOSE) e usare lo spazio occupato da questa per il nuovo input. La RENAME si trova in memoria da \$A281 a \$A297.

Le modifiche da fare, dopo essere passati al Monitor con CALL-151, sono le seguenti (tra parentesi il vecchio valore):

1) sostituire il puntatore alla RENAME con quello della CLOSE.

*9D3C:E9 (80)

2) sostituire il salto alla routine originale con uno alla nostra

*A055:81 A2 (93 A1)

3) caricare la nostra routine

*A281:AE 5D AA BD 00 02 C9 8D F0 07 E8
8E 5D AA C9 AC 60 8E 03 B2 60

(ex RENAME)

4) inizializzare un nuovo dischetto con il DOS modificato

JINIT HELLO

Il programma in Basic di figura 1 effettua automaticamente tutte queste operazioni (INIT compresa). Da notare che la modifica è compatibile con quella presentata sul numero 26 a pagina 75 che attivava un avvertimento nel caso si tentasse di salvare un programma con un nome già presente sul CATALOG. Cosa che può succedere più facilmente usando le abbreviazioni.

```

10 TEXT : HOME : PRINT : PRINT "
    PROGRAMMA CHE MODIFICA IL
    DOS "
20 PRINT " PER POTER USARE DEI N
    OMI DI FILE"
30 PRINT " ABBREVIATI.": PRINT
40 PRINT : PRINT "ATTENZIONE, LA
    RENAME E' DISABILITATA!"
50 POKE 40252,233: REM RENAME=C
    LOSE
60 POKE 41045,129: POKE 41046,16
    2: REM JSR NAME$
70 REM NAME$ MODIFICATA
80 FOR I = 41601 TO 41623
90 READ D: POKE I,D
100 NEXT
200 REM INIZIALIZZA
210 D$ = CHR$(4)
220 PRINT : PRINT " INSERISCI IL
    DISCO DA MODIFICARE "
230 PRINT : INPUT "NOME DEL FILE
    DI HELLO :";A$
240 PRINT D$"INIT"A$
250 END
500 DATA 174,93,170,189,0,2,201,
    141,240,7,232,142,93,170,201
    ,172,96,140,3,178,96,0,0
  
```

Figura 1 - Programma che modifica il DOS in modo da poter usare dei nomi abbreviati per identificare i file.

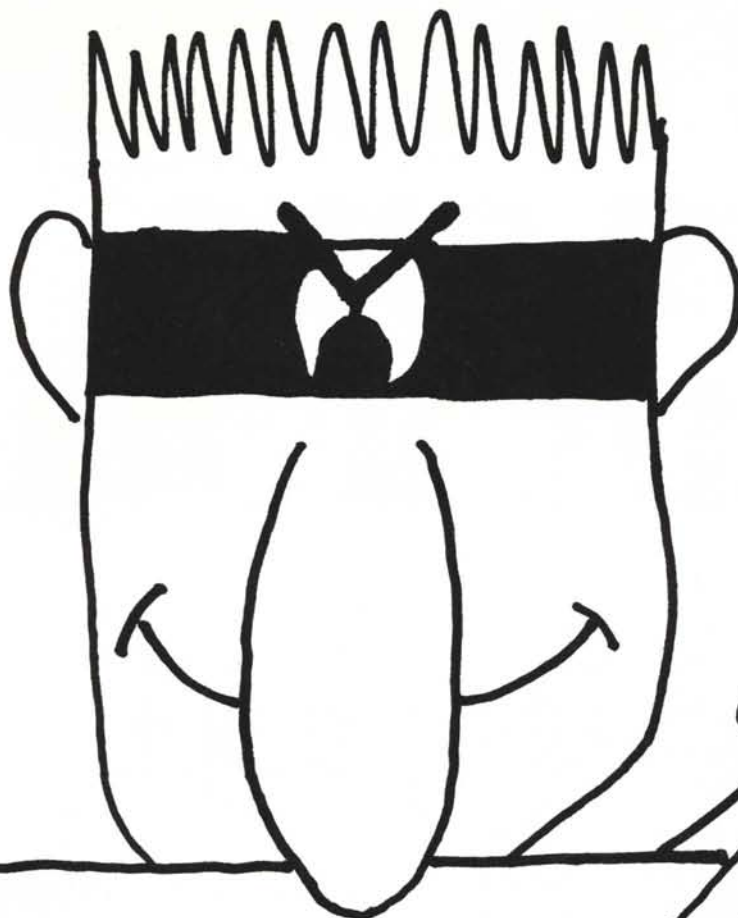
NUOVO
LISTINO
PREZZI

OLTRE 1000
VOCI ELENcate

RICHIEDETELO
ALLEGANDO
2000 LIRE
ANCHE IN
FRANCOBOLLI

OFFERTA
DEL MESE
(SOLO POCHI
PEZZI DISPON.)

CASIO FP200
NUOVO
GARANZIA
IMBALLO ORIG.
629.000
IVA COMPRESA



VIENI ALLA SUMUS!

NEL NUOVO FAVOLOSO NEGOZIO UNA SALA PROVE

TUTTA PER TE - 22 MONITOR A COLORI (TRA CUI IL NUOVISSIMO 45

POLLICI) E 22 COMPUTERS E VIDEO GIOCHI PRONTI PER ESSERE

PROVATI PRIMA DELL'ACQUISTO! UNA COSA MAI VISTA!

SUMUS: IL PIÙ GRANDE ASSORTIMENTO DI HOME COMPUTERS

VIDEO - GIOCHI - LIBRI - PROGRAMMI E ACCESSORI

VE LO DICE

SUPER SUMUS!

SUMUS

SUMUS s.r.l.
Via S. Gallo 16/r
50129 Firenze
tel. 055/29.53.61
tlx. 57.10.34

Le soluzioni non sono tutte uguali.



Qualunque sia il vostro problema da noi troverete una soluzione, la migliore! Vi offriamo tutta la gamma dei personal computer APPLE e uomini pronti a collaborare con voi giorno dopo giorno.

Fate subito quello che più di un milione di persone ha già fatto prima di voi. Scegliete la soluzione APPLE.

Venite da noi.
Scoprite la migliore.

L&L computers

L.go II Giugno, 4 - 70125 Bari
Tel. 080/224277 Telex 812082 LLBA-I

Rivenditore Autorizzato



apple computer