a cura di Fabio Marzocca

Questo mese è nuovamente di scena il PC-1500 e, fra gli innumerevoli programmi ricevuti in questo periodo, si pone prepotentemente in luce l'ottimo lavoro svolto da Ernesto de Bernardis di Catania. Oltre ad un interessante micro word-processor, de Bernardis ci ha inviato pagine e pagine manoscritte, frutto di un'estenuante ricerca sul sistema operativo del PC-1500. In questo modo sono stati ricavati ulteriori codici macchina e si avvicina così sempre più il momento in cui potremo programmare in assembler su questo maxi-pocket (o mini-personal?).

Il traguardo da raggiungere insieme, quindi, è per ora la completa stesura dei codici operativi del PC-1500 per poi dedicarci alla ben più ardua impresa di dare una risposta alla domanda: perché la Sharp non ha voluto rendere noti questi codici?

MAMMUTH
Un micro word processor

di Ernesto de Bernardis (Catania)

Un micro word processor che magari non sarà tanto utile per scrivere romanzi, ma che per qualche letterina, biglietto da visita, noterella, può rivelarsi utile e d'effetto: l'ideale sarebbe avere l'espansione di memoria da 4 a 8 K, ma anche con la configurazione di memoria minima si può fare qualcosa.

Mammuth gestisce 36 caratteri per riga, permette prima della stampa inserimenti, correzioni, cancellazioni, è equipaggiato di wrap-around (non spezza cioè le parole in fin di riga, ma le scrive tutte intere al rigo successivo), consente di stampare righe giustificate a destra, di cambiare colore, di saltare rigo, di inserire un segnale acustico in un qualunque punto del testo, e di utilizzare tutti i simboli grafici implementati nel PC-1500, eccetto la "chiocciolina" @ che costituisce un carattere speciale di controllo.

Il testo viene inserito in righe REM all'inizio del programma: è un tipo di editor piuttosto grezzo e primitivo ma è il più efficace che mi sia venuto in mente. Si consiglia vivamente di non utilizzare tutti gli 80 caratteri della linea basic, altrimenti gli inserimenti saranno alquanto problematici, con necessità di duplicare la riga REM per contenere gli ultimi caratteri della linea.

Le linee REM si devono considerare direttamente contigue, quindi una parola che non trova posto tutta intera in una linea, può essere continuata nella linea successiva, senza nessuna preoccupazione. Il ritorno a capo è gestito automaticamente dal programma in fase di stampa, quindi il testo può essere digitato in modo continuo.

Per quanto riguarda i comandi speciali, essi sono tutti costituiti da un solo simbolo preceduto dal carattere chiocciolina; possono essere distribuiti liberamente nell'ambito del testo in quanto, naturalmente, non verranno stampati. Eccone l'elenco:

@> genera un ritorno a capo forzato
@^ genera un segnale acustico di avvertimento
@N colore nero
@B colore blu
@V colore verde
@R colore rosso
@D stampa una riga giustificata a destra e va a capo
@# fine testo (obbligatorio)

Una volta scritto nella memoria il testo in linee REM, è sufficiente dare il RUN per effettuare la stampa (fig. 1)

Linguaggio macchina e suggerimenti vari

di Ernesto de Bernardis

Nella tabella seguente sono elecanti i risultati finora raggiunti dalle ricerche sui codici macchina del PC-1500:

Esadecimale	Decimale	Mnemonico
DD	221	INC R1
DF	223	DEC R1
A5	165	LD R1, (nn)
B5	181	LD R1, n
AE	174	LD (nn), R1
BA	186	JMP (nn)
B3	179	ADD R1, n
BD	189	SUB R1, n
A3	163	ADD R1, (nn)
AD	173	SUB R1, (nn)

Altri codici sono pubblicati sul n° 14 di MC. R1 rappresenta uno dei registri interni della CPU ed i codici mnemonici sono tratti dall'assembler dello Z-80. Il simbolo (nn) indica un registro (2 byte: HI,LO) mentre n indica un valore numerico da 00 a FF.

Tutto ciò almeno nei momenti di lucidità: non mancano infatti eccezioni e stranezze che invito i lettori, a spiegare. Osserviamo questi programmi in linguaggio macchina:

Figura 2 — Il programma pone nella locazione 18100 il valore 50, e nella locazione 18101 il valore (50+20)=70. Ma.....

Figura 3 — In teoria dovrebbe funzionare come il precedente: invece di partire con 50 in R1 parte con 51 e poi decrementa. In effetti nella locazione 18100 risulta 50, ma nella locazione 18101 si trova un inesplicabile 71. Ed ancora....

Figura 4 — Questo programma pone in R1 il valore 10, sottrae 8 e pone il risultato nella locazione 18100. Funziona regolarmente.

Figura 5 — Dovrebbe comportarsi come il precedente sottraendo 5 da 10: disgraziatamente risulta 15....

Gli stessi strani fenomeni avvengono col codice 163 e 173.

Ricordiamo che i programmi in linguaggio macchina possono essere caricati mediante l'istruzione:

POKE ind, C1, C2, C3....

dove ind è l'indirizzo di partenza della routine da caricare e C1, C2, C3... sono i codici della routine. Unico limite di questa potente istruzione è la lunghezza del buffer di linea di 80 caratteri: in pratica allora conviene caricare i codici a blocchi di 10, incrementando ind di 10 in 10.

La routine caricata viene eseguita con l'istruzione CALL ind, sotto forma di subroutine: per questo tutti i programmi in linguaggio macchina devono essere conclusi con il codice 154 (RETURN).

Alcuni dei codici di cui sopra sono stati scoperti osservando le routine in ROM delle funzioni CLEAR e LOCK/UNLOCK. Per trovare gli indirizzi di partenza delle routine di ogni istruzione Basic, si può utilizzare il programmino in figura 6. Questo chiede all'inizio

```
ELABORATO

                    Catania, 6/1/1983
Ernesto de Bernardis
Via Pietra dell^Ova402
95030 Trappeto (CT)

Ecco un^esempio di testo elaborato
da MAMMUTH, il micro word processor
per lo SHARP PC-1500.

SORGENTE
```

```
10:REM @NCatania,
    6/1/1983@D
20:REM Ernesto de
    Bernardis @>V
    ia Pietra dell
    ^Ova402@>
30:REM 95030 Trap
    peto (CT)@>
```

```
40:REM @>Ecco un^
    esempio di tes
    to elaborato d
    a MAMMUTH, il
    micro word pro
    cessor
50:REM per lo SH
    ARP PC-1500.@#
```

Esempio di applicazione del programma "Mammuth" con testo - sorgente e testo - elaborato.

Programma "Mammuth"

```
60000:CLEAR :CSIZE
      1:I=16586:
      DIM R$(0)*36
      ,F$(0)*36:
      WAIT 0
60010:"CARATTERI":
      IF LEN R$(0)
      =36GOTO "WRA
      P-AROUND"
60020:C=PEEK I:IF
      C=13LET I=I+
      6:GOTO 60020
60030:IF C=64GOTO
      "COMANDI"
60040:C$=CHR$ C:R$
      (0)=R$(0)+C$
      :PRINT C$;
60050:I=I+1:GOTO "
      CARATTERI"
60100:"WRAP-AROUND
      ":U=36:F$(0)
      =""
60110:D=PEEK I:IF
      D=13LET D=
      PEEK (I+6)
60120:GOSUB "PROVA
      ":IF FGOTO "
      STAMPA"
60130:"RICERCA"D=
      ASC MID$ (R$
      (0),U,1)
60140:GOSUB "PROVA
      ":IF FGOTO "
      STAMPA"
60150:U=U-1:GOTO "
      RICERCA"
60160:"PROVA"F=(D=
      46OR D=63OR
      D=33OR D=64
      OR D=45OR D=
      32OR D=44OR
      D=58OR D=59)
      :RETURN
60170:"STAMPA"F$(0
      )=RIGHT$ (R$
      (0),36-U):R$
      (0)=LEFT$ (R
      $(0),U)
60180:LPRINT R$(0)
      :R$(0)=F$(0)
      :GOTO "CARAT
      TERI"
60200:"COMANDI"I=I
      +1:S=PEEK I:
      S$=CHR$ S:IF
      S=13LET I=I+
      5:GOTO 60200
60210:IF S$="#"
      LPRINT R$(0)
      .BEEP 10,6:
      WAIT :CLS :
      PRINT "FINE
      TESTO":END
60220:IF S$=">"
      LPRINT R$(0)
      :R$(0)=""
60230:IF S$="^"
      BEEP 3,100
60240:IF S$="N"
      GOSUB "CAMBI
      A COLORE":
      COLOR 0
60250:IF S$="B"
      GOSUB "CAMBI
      A COLORE":
      COLOR 1
60260:IF S$="V"
      GOSUB "CAMBI
      A COLORE":
      COLOR 2
60270:IF S$="R"
      GOSUB "CAMBI
      A COLORE":
      COLOR 3
60280:IF S$="D"LET
      Z=36-LEN R$(
      0):FOR A=1TO
      Z:R$(0)=" "+
      R$(0):NEXT A
      :U=36:I=I+1:
      GOTO "STAMPA
      "
60500:I=I+1:GOTO "
      CARATTERI"
61000:"CAMBIA COLO
      RE"LPRINT R$
      (0);:F$(0)="
      ":LCURSOR 0:
      IF LEN R$(0)
      =0GOTO 61020
61010:FOR A=1TO
      LEN R$(0):F$
      (0)=F$(0)+"
      ":NEXT A
61020:R$(0)=F$(0):
      RETURN
```

Figura 1

```
LD R1,n      181
50            50
LD nn,R1     174
18100         70
             180
SUM R1,n     179
20            20
LD nn,R1     174
18101         70
             181
RET          154
```

Figura 2

```
LD R1,n      181
51            51
DEC R1       223
LD nn,R1     174
18100         70
             180
SUM R1,n     179
20            20
LD nn,R1     174
18101         70
             181
RET          154
```

Figura 3

```
LD R1,n      181
10            10
SUB R1,n     189
8              8
LD nn,R1     174
18100         70
             180
RET          154
```

Figura 4

```
LD R1,n      181
10            10
SUB R1,n     189
5              5
LD nn,R1     174
18100         70
             180
RET          154
```

Figura 5

l'indirizzo di partenza dell'analisi, e prosegue stampando il numero della locazione, il contenuto della locazione ed il valore ASCII corrispondente. Può essere facilmente adattato ad output sul solo display. Dopo ogni parola chiave, corrispondente ad uno statement basic, seguono 5 byte diversi: il 1° ed il 2° costituiscono il token della parola chiave, il 3° ed il 4° costituiscono il byte alto ed il byte basso dell'indirizzo di partenza della routine corrispondente alla parola chiave, il 5° solo Dio ed i progettisti del PC-1500 sanno a cosa serve! L'intera tabella dei comandi basic può essere analizzata mediante il programma di figura 7. Esso fornisce, per ogni parola chiave del basic, i valori token, l'indirizzo di partenza della routine, ed il 5° byte. Utilizzando questo programma è stato possibile scoprire una nuova istruzione: si tratta di una P seguita da 4 spazi, inserita nelle locazioni fra 49748 e 49752. Risulta comunque impossibile inserire questa istruzione direttamente da tastiera: procediamo perciò come segue.

Eseguite il NEW e quindi scrivete la linea

100 REM

A questo punto digitate:

POKE 16585, 163 ENTER

In questo modo i token del REM (241,171) vengono trasformati nei token dell'istruzione P (241,163). Si potrà constatare a questo punto che nella linea 100 si troverà la misteriosa istruzione. Vediamo ora in alcuni punti qualche suggerimento interessante e, soprattutto, alcune precisazioni non contenute sul manuale.

1) Il nome di una variabile può essere formato da quanti caratteri si vuole (ma solo i primi due sono significativi): l'importante è che, nel corpo della parola variabile, non siano contenute sequenze di lettere costituenti parole-chiave del basic: per esempio sono accettabili le variabili MODO e PUNTEGGIO mentre non lo sono POSIZIONE e SINISTRA le quali contengono le parole chiave ON e SIN.

2) È possibile, nell'ambito di un programma, incorporare una linea REM incancellabile, che può contenere nome e cognome del programmatore, titolo e commenti del programma etc. È conveniente incorporare questa o queste linee verso la metà del programma rendendo così la vita difficile e chi le voglia cancellare: si tratta infatti di un gioco di byte e chi volesse cancellare o laterare la nostra riga-copyright dovrebbe trovare le locazioni ove è posto il suo numero, un compito evidentemente poco simpatico da effettuare in un mare di byte.

Osservate il programma di figura 8. Volendo rendere incancellabile la riga 20, dovremo operare come segue:

a) battere tutto lo spezzone di programma precedente la riga da rendere incancellabile.

b) Digitare STATUS 2 - 1 ed annotare il valore fornito dal computer.

c) Scrivere tutta la riga REM con il suo regolare numero di linea

d) Sia x il numero che abbiamo annotato nella fase b; digitare POKE x, 0,0. Questo porta a 0 il numero di riga della REM.

e) Continuare a battere regolarmente il resto del listato.

f) Inserire, nella riga precedente il REM incancellabile, un'istruzione GOTO che indirizzi alla linea dopo il REM: questo perché il sistema operativo, incontrando una linea basic con numero di riga = 0 segnalerebbe un ERROR 1.

Il manuale non vi accenna, ma il nostro Sharp possiede una delle caratteristiche più celebrate di grossi calcolatori come l'HP 87, o di piccoli mostriciattoli factotum come l'HP 41C. Questa caratteristica è la completa etichettabilità, con etichetta lunga fino a 73 caratteri alfanumerici, maiuscoli, minuscoli e grafici. Sono perfettamente leciti quindi statement del tipo RUN "OROLOGIO", GOTO "Fine programma", ON X GOSUB 250, "T", 100, "TASSE".

Si potrebbe perfino scrivere i programmi a prescindere dal numero di riga, risolvendo tutti i GOTO e GOSUB con etichette, e guadagnando in chiarezza e in possibilità di rinumerazione.

4) Il programma di figura 9, senza la linea 45, funge da cronometro: parte con RUN, si ferma con BREAK ed in T contiene i secondi trascorsi. Se aggiungete la linea 45 e sostituite agli asterischi il tempo desiderato, otterrete un buon timer che vi avvertirà con un gentile ma fermo BIP BIP. Il programma è volutamente grezzo e poco curato, perché destinato a far parte, magari come subroutine, di programmi più completi e complessi. MC

```
10:TEXT :CSIZE 2:
   LPRINT "ANALIS
   I BASIC":CSIZE
   1
20:LPRINT "ISTRUZ
   IONE   TOKEN
      IND     BYTE
    5":LF 1
30:A=49237
40:A$=""
50:C=PEEK A:IF C=
   240OR C=241
   GOTO 100
60:A$=A$+CHR$ C:A
   =A+1:GOTO 50
100:T1=PEEK A:T2=
   PEEK (A+1):I=2
   56*PEEK (A+2)+
   PEEK (A+3):B=
   PEEK (A+4)
110:LPRINT USING "
   &&&&&&&&&&";A$
   ;USING "####";
   T1;T2;USING "#
   #######";I;TAB
   28;B
120:A=A+5:GOTO 40
```

Figura 7 - Programma per la ricerca dei Token e degli indirizzi delle parole-chiave.

```
10:INPUT "IND. BA
   SE? ";A:USING
   "######"
20:LPRINT A;PEEK
   A;"    ";CHR$
   PEEK A
30:A=A+1:GOTO 20
```

Figura 6 - Routine per l'analisi del contenuto della ROM.

```
10:BEEP 5,6
20:REM Ernesto
30:PRINT "CIAO"
```

Figura 8 - Inserzione di una riga incancellabile.

```
10:BEEP 5,6:GOTO
   30
 0:REM Ernesto
30:PRINT "CIAO"
```

```
10:T=0
20:Q=TIME
30:IF TIME =QGOTO
   30
40:T=T+1
50:GOTO 20

45:IF T=***BEEP 2
   ,0,6:END
```

Figura 9 - Contasecondi e timer.

Una precisazione sul Renumber

Sul numero 14 di MC, abbiamo pubblicato il programma di RENUMBER per PC-1500 inviatoci dal lettore Luca Ridarelli. Nel testo dell'articolo non è stata citata in modo esplicito una importante limitazione del programma: il renumber infatti non rinumera gli indirizzi delle GOSUB, GOTO, THEN e RESTORE.

Si consiglia perciò di etichettare questi statement ed effettuare così l'indirizzamento implicito. Chiediamo scusa per la distrazione e ringraziamo i lettori che ci hanno inviato la segnalazione.

Desideriamo comunque sottolineare che programmi privi di questa limitazione sono, sì, possibili da realizzare, ma risultano a nostro avviso troppo ponderosi per essere mantenuti in memoria insieme al programma principale in una macchina come la PC-1500.