i trucchi del CP/M

a cura di Claudio Rosazza

BASIC ASSEMBLER - IIIª parte

Come già accennato nella precedente puntata questo mese affronteremo il problema del recupero dell'area di memoria occupata dal CCP del CP/M al fine di poterla utilizzare come Work-area per il Basic.

In effetti, il Basic all'atto del caricamento in Ram, non considera l'area del CCP come area riservata di sistema e la usa tranquillamente come Work-area, ma nel nostro caso, considerato che deve coesistere anche una subroutine in Assembler, i problemi si complicano.

Nella scorsa puntata, infatti, avevamo adottato la soluzione di allocare la subroutine Assembler appena al disotto del CCP ed il limite superiore della Work-area del Basic appena al di sotto dell'inizio della subroutine.

Non possiamo allocare la subroutine Assembler nell'area del CCP perché nel momento in cui il programma di caricamento in Ram della subroutine restituisce il controllo al CP/M per dare la possibilità all'utente di caricare il Basic, il sistema operativo opera d'ufficio un warm-boot che comprende fra le altre cose anche un caricamento del CCP con conseguente distruzione della subroutine precedentemente caricata.

Il trucco per poter utilizzare l'area del CCP consiste nello spostare la subroutine Assembler all'interno della Ram in due fasi successive; la prima attraverso un ciclo di rilocazione insito nella subroutine stessa e la seconda attraverso un ciclo di rilocazione contenuto nella subroutine, ma questa volta richiamato dall'interno del Basic.

In figura 1 troviamo la situazione della Ram dopo aver effettuato il bootstrap del CP/M con la consolle in A > pronta ad accettare un qualsiasi comando.

Ipotizziamo quindi di caricare la nostra subroutine Assembler che si andrà automaticamente a rilocare in un'area di Ram che deve rispondere a due particolari esigenze.

Deve essere infatti compresa fra il limite superiore occupato dall'interprete Basic, perché successivamente dovremo caricare l'interprete senza distruggere la subroutine, e il limite inferiore del CCP poiché nel ritorno al CP/M il restore del CCP stesso non deve alterare la subroutine appena caricata.

In figura 2 possiamo osservare la situazione della Ram dopo il caricamento della subroutine e la successiva rilocazione intermedia alla locazione 9000H (esempio).

A questo punto la subroutine restituisce il controllo al sistema operativo e ci permette di caricare l'interprete Basic con l'opzione /M:&H8FFF in modo da istruire il Basic a vedere la fine della propria Work-area appena al di sotto della prima rilocazione della subroutine Assembler (Fig. 3).

Dall'interno del Basic effettuando una CALL alla locazione 9000H provocheremo la seconda rilocazione della subroutine Assembler nell'area dedicata al CCP avendo cura di non oltrepassare il TPA effettivo del calcolatore contenuto come informazione nelle locazioni 6 e 7 (Fig. 4).

Operando quindi una istruzione di CLEAR dall'interno del Basic si sposterà il TPA interno del Basic fermo a 8FFFH appena al disotto dell'inizio della subroutine dopo la seconda rilocazione (Fig. 4). Appare ora evidente che bypassando tutti i problemi sorti la scorsa puntata siamo riusciti ad utilizzare a pieno tutta la Ram disponibile.

Per operare la seconda rilocazione in modo automatico è possibile utilizzare il programma qui di seguito, denominato INIT. BAS, che potrà essere richiamato assieme all'interprete.

```
10 INIT = $H9000
20 CALL INIT
30 CLEAR, $HXXXX
40 X = FRE (O):X$ = MID$(STR$ (X),2,5)
:PRINT X$; " Bytes free"
50 END
```

Al posto di XXXX va inserito il valore della locazione di inizio della subroutine Assembler BEGIN-1 desumibile dal file SUB.PRN dopo aver assemblato il file SUB.MAC.

Il valore di questa variabile va calcolato in modo tale che l'ultima locazione della subroutine assembler sia al di sotto del TPA effettivo del calcolatore desumibili dalle locazioni 6 e 7 come descritto nella precedente puntata.

Il programma effettua dapprima una CALL alla locazione INIT ove viene operata la seconda rilocazione della subroutine Assembler, quindi sposta con l'istruzione CLEAR il TPA interno del Basic tale da allargare la Work-area-basic al massimo delle possibilità. Infine calcola l'ampiezza della Work-area-basic e ne scrive il valore

Figura 1

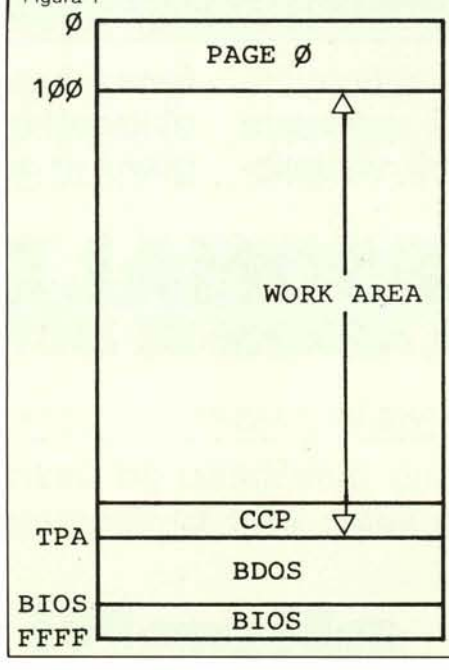
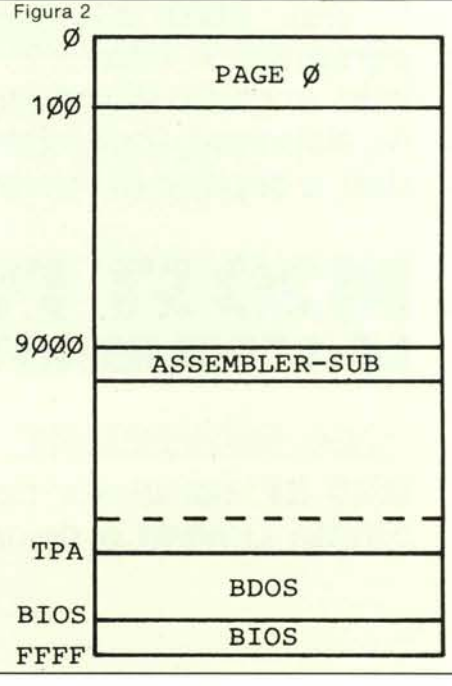


Figura 2



```
          .Z80
          ASEG
          ORG       1ØØH
;
CONIO     EQU       6
LIST      EQU       5
BDOS      EQU       5
;
BEGIN     EQU       ØAØØØ   (da controllare in
                             funzione del parti-
                             colare tipo di cal-
                             colatore)
;
BLKTRF:   LD        HL,1OEH
          LD        DE,9ØØØH
          LD        BC,JEX-ZUBØ+16
          LDIR
          JP        Ø
;
INIT:     LD        HL,9ØØCH
          LD        DE,BEGIN
          LD        BC,JEX-ZUBØ
          LDIR
          RET
;
          .PHASE    BEGIN
;
ZUBØ:     LD        E,(HL)
          LD        C,CONIO
          CALL      BDOS
          RET
;
ZUB1:     PUSH      HL
ZUB1B:    LD        E,ØFFH
          LD        C,CONIO
          CALL      BDOS
          CP        Ø
          JR        Z,ZUB1B
          LD        E,Ø
          LD        D,A
          POP       HL
          LD        (HL),D
          INC       HL
          LD        (HL),E
          RET
;
ZUB2:     LD        E,(HL)
          LD        C,LIST
          CALL      BDOS
          RET
;
JEX       EQU
;
          END
```

Figura 3

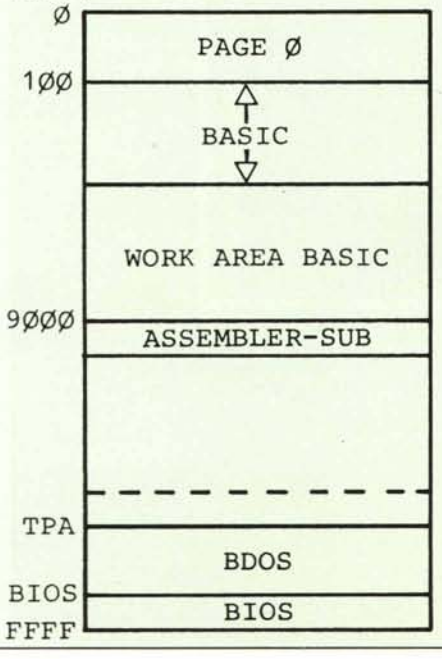
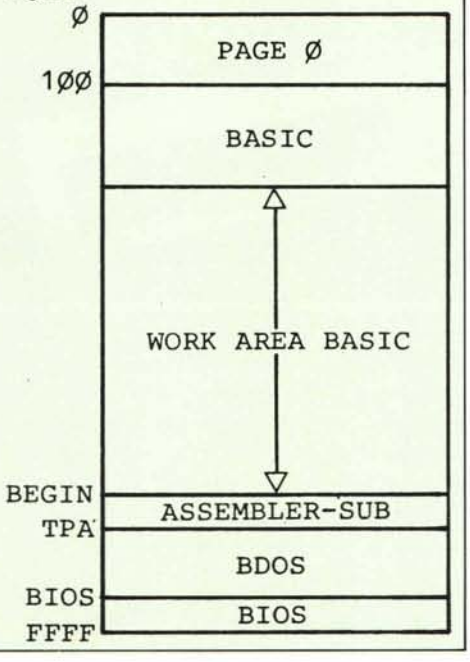


Figura 4



con lo stesso formato del prompt iniziale dell'interprete.

A questo punto definendo all'interno del programma Basic che vogliamo eseguire le locazioni assolute di ZUBØ, 1 e 2 possiamo effettuare tranquillamente le Call alle subroutine da noi definite nel programma Assembler.

La subroutine Assembler va assemblata e linkata nel modo solito come descritto nella precedente puntata; ciò che si dovrà utilizzare dopo tali operazioni sarà il file SUB.COM che contiene le subroutine e le routine di rilocazione.

Ricapitolando, per entrare all'interno del Basic avendo operative le subroutine Assembler occorre eseguire i seguenti comandi:

A>SUB<return>
A>MBASIC INIT /M: H8FFF
<return>

Occorre tenere presente che tale sequenza di comandi va data ogniqualvolta si torni in ambiente CP/M poiché il restore del CCP distrugge le subroutine Assembler allocate per l'appunto nell'area Ram del CCP. MC

Ti occorre un personal computer o un sistema multiterminale?
Se vuoi l'uno senza rinunciare all'altro...

grappolo

IL MULTI PERSONAL

Con Grappolo puoi iniziare con un personal, tutto tuo, per arrivare al Multipersonal con otto posti di lavoro indipendenti, ciascuno con 64K di memoria e unità centrale proprie, collegati via bus veloce ad una base dati comune. Con Grappolo è già disponibile una vasta biblioteca di programmi pronti all'uso, CP/M compatibili!
Grappolo, l'efficienza di un sistema distribuito con l'individualità del personal computer. Grappolo, il Multipersonal, costruito e garantito in Italia dalla lunga esperienza SAICO.

saico

SOCIETA' AZIONARIA ITALIANA COMPUTERS

20121 MILANO - Via S. Giovanni sul Muro, 1 - Tel. (02) 3452116 • 00199 ROMA - Via Asmara, 58 - Tel. (06) 8310063 • 80146 NAPOLI - Via Ferrante Imparato, 35 - Tel. (081) 7523744 • 95123 CATANIA - Via A. De Cosmi, 5 - Tel. (095) 326356