



i trucchi del CP/M

a cura di Claudio Rosazza

Buffer di tastiera

Una delle caratteristiche allettanti che vengono coinvolte nella scelta di un elaboratore operante con il sistema operativo CP/M è sicuramente data dalla presenza del buffer di tastiera.

Tale particolarità consente al calcolatore di accettare caratteri digitati da tastiera anche quando il microprocessore sta effettuando tutt'altro lavoro; questi caratteri vengono memorizzati in una particolare zona di Ram organizzata con una struttura FIFO circolare. In pratica immaginate un tubo chiuso su se stesso a forma di cerchio con un ingresso ed una uscita verso l'esterno che possono scorrere lungo la circonferenza. Inizialmente ingresso ed uscita sono nello stesso punto. La struttura FIFO viene caricata dal canale di ingresso e lo stesso viene avanzato di una posizione; se il programma ricevente è pronto per accettare il carattere, verrà avanzato anche il canale di uscita e prelevato il carattere. Viceversa, se il programma non può accettare caratteri in quel momento, il canale di ingresso alla FIFO avanzerà di una posizione per ogni carattere premuto e verrà raggiunto successivamente quando il programma ricevente prelevando dalla FIFO un carattere per volta farà avanzare il canale di uscita.

La funzione di buffer di tastiera è una di quelle tipiche operazioni che sfruttano, in genere, le capacità di gestione di interrupt da parte del microprocessore e cioè la capacità di poter interrompere momentaneamente un processo in corso per poter permettere, per esempio, l'acquisizione del carattere digitato da tastiera e la sua memorizzazione nella FIFO; il tutto avviene in modo assolutamente trasparente per il processo in corso.

L'utilità del buffer di tastiera si apprezza maggiormente quando in una procedura applicativa vi sono dei successivi input di dati intercalati con calcoli intermedi che vengono svolti in una certa entità di tempo. In questo caso se l'operatore è veloce e già conosce i dati da digitare, con il buffer di tastiera non sarà costretto ad aspettare ogni volta la richiesta di input da parte del calcolatore, ma potrà digitare, ovviamente alla cieca (cioè senza eco immediata su video) tutti i dati in suo possesso essendo certo che verranno recuperati dal programma non appena accederà alla FIFO di cui sopra.

L'utente che acquisterà un calcolatore con sistema operativo CP/M e che intende lavorare in interprete Basic Microsoft avrà, però, una amara sorpresa. In ambiente interprete Basic il buffer di tastiera non funziona! O meglio, il buffer funziona, ma viene ingiustamente scaricato dal Basic stesso.

Il motivo è da ricercare nella funzione di Break dell'interprete che consente di interrompere un programma in qualsiasi momento per poi riprenderlo subito dopo. Tale funzione viene realizzata con la digitazione del carattere Control-C che causa l'interruzione e con l'istruzione CONT che riprende l'esecuzione del programma.

Per dare la possibilità all'operatore di inter-

rompere il programma in qualsiasi momento il Basic è costretto dopo aver eseguito una istruzione e per tutte le istruzioni contenute nel programma a fare una chiamata al sistema operativo per sapere se vi è un carattere pendente da tastiera; nel caso di risposta affermativa il Basic è costretto a leggerlo per verificare se è un C. In caso di C viene eseguito il Break del programma altrimenti quel carattere viene perduto ed il Basic prosegue ad interrogare il CP/M sullo stato della tastiera fermandosi solo quando non vi sono più caratteri pendenti o vi sia un C.

Il trucco per far funzionare il buffer di tastiera anche sotto interprete Basic consiste nel disabilitare la chiamata di stato tastiera al CP/M. L'operazione consiste nella modifica di 3 byte all'interno del Basic. Purtroppo tale modifica non è possibile effettuarla in modo permanente come il Print Using (MC n. 14) né si può effettuare

```

*
      .Z80
      ASEG
      ORG      100H
*
BASCAL EQU      12CEH
*
BEGIN  EQU      8000H
*
*
BLKTRF: LD      HL, 10EH
        LD      DE, BEGIN
        LD      BC, JEX-ONBUF
        *
        LDIR
        JP      0
*
*
      .PHASE  BEGIN
*
ONBUF: LD      HL, BASCAL
        LD      A, 3EH
        LD      (HL), A
        LD      A, 0
        INC    HL
        LD      (HL), A
        INC    HL
        LD      (HL), A
        RET
*
JEX    EQU      $
*
      END

```

Listato della subroutine Assembler per l'abilitazione del buffer di tastiera in ambiente interprete Basic Microsoft.

dall'interno del Basic tramite dei Poke; risulta per forza necessario utilizzare una subroutine esterna in Assembler.

La subroutine, riportata in questa pagina, va scritta usando un qualsiasi editor per programmi e salvata col nome di BUFFER.MAC.

La variabile BASCAL va adattata alla versione di Basic che si intende utilizzare: per il 4.51 esso vale 1247H, per il 5.1 vale 12A8H e per il 5.2 vale 12CEH.

Il programma va assemblato e linkato con i seguenti comandi:

A>M80 BUFFER = BUFFER <return>

A>L80 BUFFER, BUFFER/N/E <return>

Il prodotto sarà un programma di nome BUFFER.COM. Tale programma se richiamato provvederà a sistemare in una determinata zona di RAM la subroutine richiamabile dal Basic per la disabilitazione della chiamata di stato al CP/M. Quindi, per entrare in Basic occorre digitare i seguenti comandi:

A>BUFFER <return>

A>MBASIC /M:&H7FFF <return>

L'interprete verrà caricato e fornirà ben pochi byte liberi, ma non preoccupatevi! (Solo per poco). Prima di eseguire qualsiasi altro comando o funzione è necessario far girare il programma qui di seguito:

10 ONBUF = & H8000

20 CALL ONBUF

30 TPA = PEEK (7) * 256 + PEEK (6)

40 CLEAR, TPA-1

50 X = FRE (0):X\$ = MID\$(STR\$(X), 2,5)

60 PRINT X\$; "Bytes Free"

Questo programma effettua la Call alla subroutine Assembler che disabilitando la chiamata di stato al CP/M abilita la gestione del buffer di tastiera anche in interprete Basic e successivamente ripristina il TPA interno del Basic restituendo tutta la memoria disponibile all'interprete.

A questo punto occorre fare due precisazioni. La prima riguarda il funzionamento del C che viene ad essere limitato, per forza di cose, solo quando il programma si trova in una fase di input dati, la seconda, invece, riguarda la necessità in alcuni casi, di dover scaricare volontariamente il buffer di tastiera, per esempio, dopo che un controllo dimensionale su un dato di ingresso ha dato esito negativo e quindi deve essere richiesta la ridigitazione del dato.

Una tale funzione di scaricamento del Buffer si può facilmente operare usando l'istruzione INKEY\$ come di seguito:

100 Z\$ = INKEY\$

110 IF Z\$ = "" THEN RETURN

120 GOTO 100

Effettuando un Gosub 100 la subroutine ritornerà solo dopo aver scaricato l'intero buffer di tastiera.

Va infine considerato il fatto che se uscite dal Basic e volete rientrarvi mantenendo il buffer di tastiera attivo, potete farlo solo con un Recovery (MC n. 13) o ripetendo l'intera operazione di cui sopra.

Vi consigliamo di usare l'interprete con il buffer di tastiera attivo solo con programmi già messi a punto poiché la funzione di Break operata con il C, che viene a mancare con il buffer di tastiera attivo, è un utile strumento di debug.

Errata corrige

MCmicrocomputer N. 19 "I Trucchi del CP/M"

Nel listato Assembler alla terza riga della routine BLKTRF l'istruzione LD HL, JEX-ZUB0 va sostituita con LD BC, JEX-ZUB0.

MCmicrocomputer N. 20 "I Trucchi del CP/M"

Nel listato del programma INIT.BAS alle righe 10 e 30 il dollaro va sostituito con &. Nell'ultima colonna a pagina 101 il comando MBASIC INIT/M: H8FFF manca della & prima della H.

QUOTAZIONI

Materiale nuovo imballato

**CENTRO
ASSISTENZA
SPECTRUM**

SUMUS

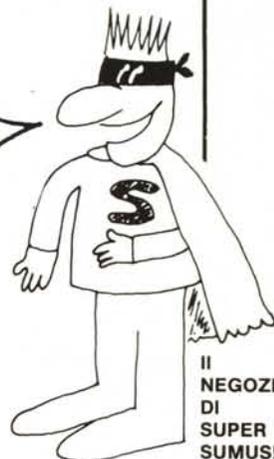
SUMUS s.r.l.
Via S. Gallo 16/r
50129 Firenze
tel. 055/29.53.61
tlx. 57.10.34

16K L. 389.000 IVA INC.

AL PARADISO DEI

SINCLAIR ZX SPECTRUM

**IL PIÙ GRANDE ASSORTIMENTO
ITALIANO DI ACCESSORI!**



**IL
NEGOZIO
DI
SUPER
SUMUS!**

Tutto per SPECTRUM:

Amplificatore Audio	18.300 IVA inc.
Generatore di suoni programmabile	52.500 IVA inc.
"Orator" Sintet Vocale	105.000 IVA inc.
Master Unit (contiene Sound Synth., Ampl. Audio, Orator, Interf. X Joystick)	144.000 IVA inc.
Interfaccia RS 232	91.500 IVA inc.
Interfaccia Centronics	91.500 IVA inc.

Tastiera/contenitore per
SPECTRUM o ZX-81.

Finalmente potrete usare comodamente
il vostro microcomputer!
L. 79.000 IVA inc.



Mille altre novità, altri computers, video giochi,
programmi ecc. ecc. Visitateci!



Grandioso assortimento di libri per SPECTRUM
novità del mese (in inglese)

L'hardware dello SPECTRUM.

Come conoscere ogni dettaglio.

Come costruire facilmente una tastiera
ausiliaria - il Joystick - l'Interf. stampante - le
Interfacce AD ecc. ecc.

Tutta la ROM SPECTRUM disassemblata
istruzione per istruzione con spiegazioni.

20 giochi per lo SPECTRUM

Disponibili: Editor/Assembler - Debug - Forth.

Si cercano concessionari

PIU' DOMANDE PIU' RISPOSTE

Olivetti M20 personal computer: pronto per ogni impiego tecnico-scientifico. Naturalmente potete utiliz-

zarlo per le applicazioni più semplici, ma per apprezzarlo del tutto dovrete consultarlo intorno a questioni complesse, chiedendogli di rendere possibili decisioni efficaci e di contribuire a prevedere, progettare e programmare. Chiedetegli di più e otterrete di più. M20 personale e riservato, in

grado di produrre, accumulare, elaborare, trasmettere e archiviare dati, e capace di riutilizzarli, visualizzarli

simultaneamente e stamparli, producendo dattiloscritti, tabulati, grafici e disegni. M20: potente come può

esserlo un computer a 16 bit, e di magnifico disegno, unità video orientabile e separabile a 12 pollici ed a 8 colori, dotato di diversi linguaggi e di ampie possibilità di collegamento con periferiche e strumenti esterni.

M20: dall'azienda che si pone in modo innovativo nell'elettronica dell'in-

formazione ed offre strumenti immediatamente efficaci, ma pronti ad integrarsi in seguito con altri.



Anche in leasing con Olivetti Leasing S.p.A.

M20: PERSONAL COMPUTER **LO SCEGLIERESTE ANCHE SE NON FOSSE OLIVETTI**

olivetti

M20, distribuito ed assistito in tutta Italia da una vasta rete di concessionari e rivenditori. Consultate gli elenchi telefonici.