VIC da zero +64

di Tommaso Pantuso

Un file per il video (seconda parte)

La volta scorsa abbiamo imparato ad utilizzare un file sequenziale in lettura e in scrittura. Abbiamo visto come sia possibile, servendosi di esso, memorizzare delle ingenti moli di dati, sotto forma di record, su una memoria di massa, quale potrebbe essere il floppy disk, e come rileggerle.

Oggi ci serviremo, come promesso, di un file di questo tipo, inglobato in un programmino dimostrativo, che ci permetterà di gestire, su disco, un'intera videata.

Cosa vogliamo fare

Cominciamo col dire che per linearità di esposizione durante il discorso ci riferiremo ad una sola macchina, cioè al Commodore 64, proponendoci di dare, alla fine dell'articolo, tutte le informazioni necessarie alla conversione su Vic 20 dei concetti esposti.

Il programma di editor che vi proponiamo è tutto in Basic e quindi di immediata interpretazione. I suoi scopi sono puramente dimostrativi e con esso ci proponiamo di riassumere buona parte delle nozioni trattate nel corso degli ultimi articoli. Siamo sicuri che i lettori più attenti sapranno farne buon uso traendo da esso uno spunto per realizzare programmi più complessi e di maggiore utilità. Detto ciò, non ci resta che passare a descrivere il funzionamento dell'utility in questione esaminando per prima cosa i blocchi da cui è composta. Essi sono rappresentati, in maniera sintetica, nella figura 1 dove troviamo indicate le funzioni svolte da ciascun blocco del listato della figura 2, ognuno dei quali è accessibile premendo uno dei tasti funzione indicato in alto. Con f1, il contenuto della memoria di schermo viene posto in una zona della memoria che abbiamo precedentemente protetto da intrusioni spostando gli opportuni puntatori (linea 3). Questa opzione è stata prevista per mantenere in memoria Ram una schermata anche quando essa non compare più sullo schermo. Se ad esempio vogliamo modificare in più modi una videata e conservarla ogni volta su disco, servendoci del buffer creato possiamo partire, per le modifiche, sempre dallo schermo originale evitando di ricomporlo tutte le volte. Per spiegarci meglio, potremmo aver bisogno di creare delle tabelle contenenti varie voci e memorizzare ciascuna delle schede così ottenuta su disco. Possiamo allora creare il disegno della scheda di base una sola volta e memorizzarlo nel buffer richiamandolo ogni volta che si vuol effettuare una nuova compilazione: è evidente il notevole risparmio di tempo che si ottiene.

Dato che, come già detto, il programma che vi forniamo è puramente dimostrativo, noi abbiamo previsto un buffer che può contenere un solo schermo ma, naturalmente, voi potete estendere la zona di memoria preposta a questo scopo per ottenere la memorizzazione di più videate richiamabili singolarmente a seconda delle esigenze. Le linee di programma che svolgono la funzione che stiamo descrivendo sono quelle che vanno da 200 a 240 e su esse riteniamo inutili ulteriori commenti.

Una volta che lo schermo si trova nel buffer, può essere richiamato, sovrapponendosi ad una eventuale schermata già presente, premendo il tasto f3. L'operazione materiale viene compiuta nel segmento contenuto tra la linea 500 e la linea 550. Come è facile osservare, la prima operazione compiuta è quella che definisce tutta la zona colore, memorizzandovi il codice 1, rendendo così visibile il disegno. È ovvio che, se comporrete un programma che prevede più buffer, ciascuno di essi dovrà essere richiamato servendosi di una chiave diversa (per chiave intendiamo il tasto o i tasti da premere per effettuare la selezione).

Dopo la descrizione di queste operazioni di contorno, passiamo ad illustrare quelle più importanti di salvataggio e caricamento da disco.

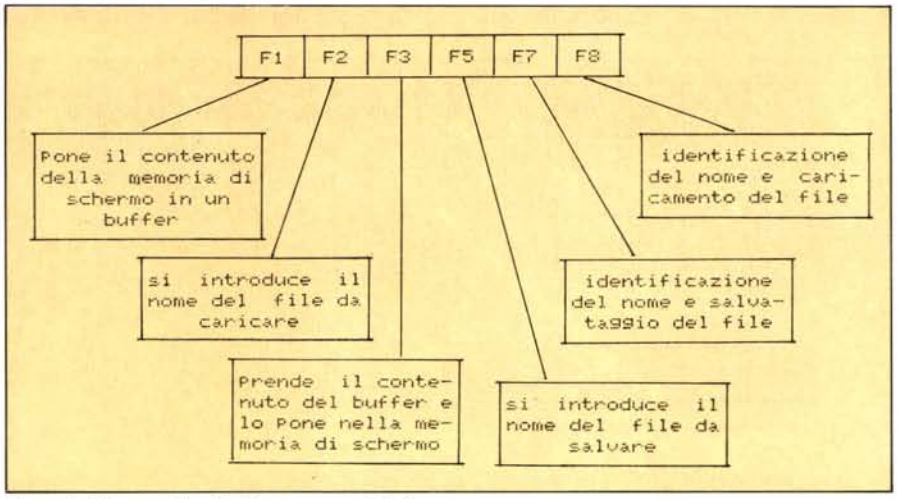


Figura 1 - Schema a blocchi del programma della figura 2.

```
0 REM --- PROGRAMMA PER LA GESTIONE DI SCHERMATE ---
1 REM --- SU DISCO PER C 64                       ---
2 REM ---------------------------------------------
3 POKE52,156:POKE56,156:POKE650,128
4 REM ---------------------------------------------
5 PRINT"□";
6 POKE53280,0:POKE53281,0:PRINT"■";
10 GETA$:IFA$=""THENGOSUB100:GOTO10
20 IFA$=CHR$(13)THENPOKE214,23:
30 IFA$=CHR$(34)THEN10
40 IFA$="■"THEN GOSUB 200
50 IFA$="■"THENGOSUB500
70 IFA$="■"THENGOTO 900
80 IFA$="■"THENGOSUB1000
85 IFA$="■"THENGOTO3000
87 IFA$="■"THENGOSUB3500
90 PRINTA$;:GOTO10
98 END
99 REM ---------------------------------------------
100 PRINT"■ ■";
110 FORI=1TO50:NEXT
120 PRINT"■■ ";
130 REM FORI=1TO100:NEXT
140 PRINT"■■";
150 RETURN
151 REM ---------------------------------------------
200 FORI=0TO959
210 POKE39936+I,PEEK(1024+I)
230 NEXT
240 RETURN
241 REM ---------------------------------------------
500 FORI=55296TO56295
510 POKEI,1:NEXT
511 REM ---------------------------------------------
520 FORI=0TO959
530 POKE1024+I,PEEK(39936+I)
540 NEXT
550 RETURN
551 REM ---------------------------------------------
900 POKE214,23:PRINT:PRINT"SAVE FILE (MAX 4 CAR.)";:GOTO10
1000 B$="":FORI=0TO3
1010 Z$=CHR$(PEEK(2006+I)+64):B$=B$+Z$
1020 NEXT:
1021 REM ---------------------------------------------
2000 OPEN7,8,7,"@0:"+B$+",S,W"
2010 FORI=1024TO1983
2020 C$=STR$(PEEK(I))
2030 PRINT#7,C$
2040 NEXT
2050 CLOSE7:RETURN
2051 REM ---------------------------------------------
3000 POKE214,23:PRINT:PRINT"LOAD FILE (MAX 4 CAR.)";:GOTO10
3001 REM ---------------------------------------------
3500 N$="":FORI=0TO3
3510 Z$=CHR$(PEEK(2006+I)+64):N$=N$+Z$
3520 NEXT:PRINT"□";
3521 REM ---------------------------------------------
4000 OPEN7,8,7,"0:"+N$+",S,R"
4005 FORI=55296TO56255:POKEI,1:NEXT
4010 FORI=0TO959
4020 INPUT#7,C$
4030 POKE1024+I,VAL(C$)
4040 NEXT
4050 CLOSE7:RETURN
```

Figura 2 - Editor di schermate.

Gestione su disco

Una volta composta una schermata, dovremo procedere all'operazione di salvataggio su disco. Questa fase, a titolo riassuntivo di alcuni importanti concetti esposti sulla memoria video, l'abbiamo suddivisa in due parti: introduzione del nome del file da registrare (riprendere da disco) e salvataggio (caricamento) vero e proprio. Passiamo a descrivere la fase di salvataggio.

Intanto è bene dire che l'ultima linea video viene da noi impiegata per la scrittura di informazioni, utili al caricamento ed al salvataggio, per cui essa non va considerata durante la composizione della videata.

Premendo f5, si passa al modo caricamento ed il programma ci chiede il nome del file da creare. Il metodo che abbiamo utilizzato per l'identificazione del nome scritto in un certo punto sullo schermo è abbastanza originale. Il nome, che per semplicità abbiamo previsto essere di massimo quattro caratteri — ciascuno dei quali rappresentato da una lettera dell'alfabeto — viene scritto in basso a sinistra sullo schermo. Ora, dato che, come sapete, qualunque simbolo che compare sul teleschermo ha un'immagine in memoria video, nelle locazioni 2006, 2007, 2008 e 2009 di tale zona troveremo i codici dei caratteri che compongono il nome stesso, naturalmente in codice di schermo. Il segmento che va dalla linea 1000 alla 1020 intercetta questi caratteri, trasforma il codice video di ognuno in codice Ascii, li unisce e ricompone il nome che abbiamo introdotto.

Se ad esempio introduciamo il nome

ABCD,

nella memoria di schermo, a partire dalla locazione 2006, avremo la seguente configurazione:

Loc.	cont.
2006	1
2007	2
2008	3
2009	4

Dato che il contenuto di ciascuna locazione codifica (in codice di schermo) una lettera dell'alfabeto, può essere convertito in codice Ascii aggiungendovi 64. Unendo infine i caratteri Ascii relativi a ciascuna locazione, avremo effettuato un processo di ricostruzione della parola "ABCD" riassumibile nella seguente tabellina e nella figura 3:

codice schermo	codice Ascii	Chr$
1	65	A
2	66	B
3	67	C
4	68	D

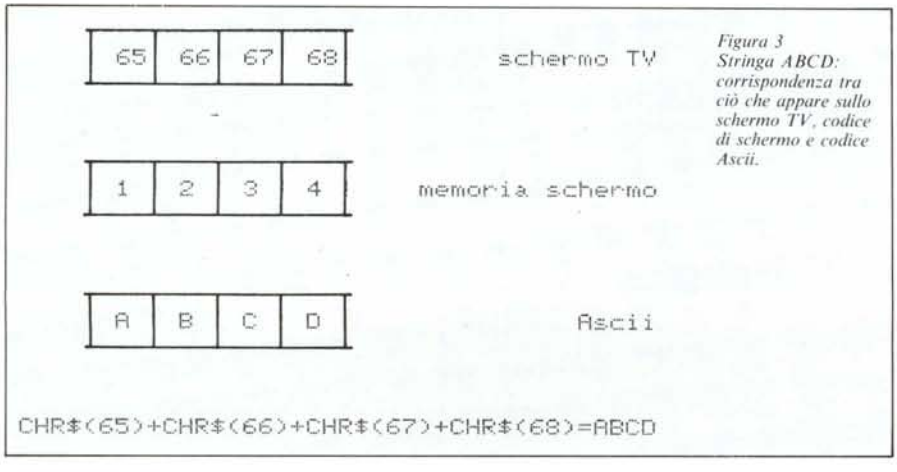


Figura 3 Stringa ABCD: corrispondenza tra ciò che appare sullo schermo TV, codice di schermo e codice Ascii.

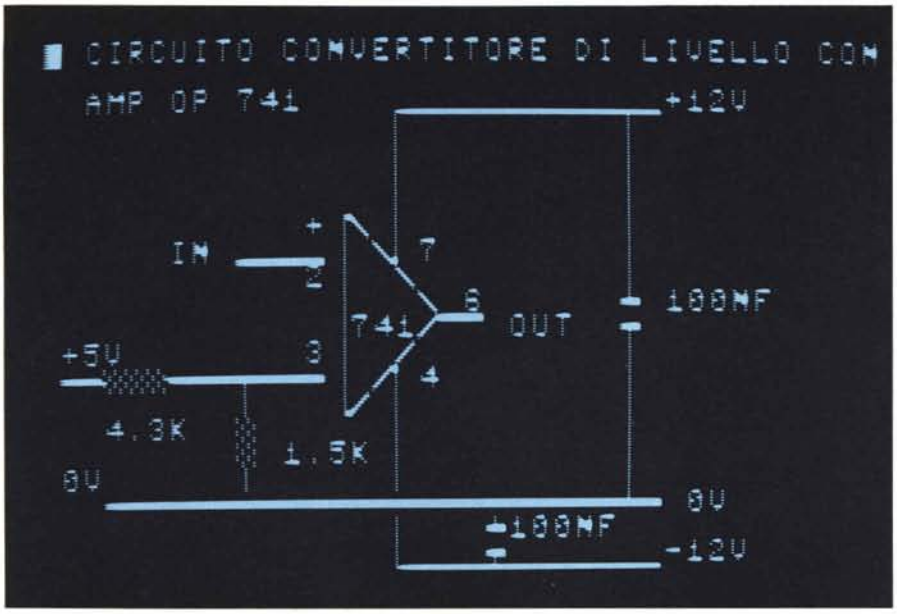
Tenete presente che la lunghezza del nome può essere variata a piacimento cambiando il ciclo For... Next della linea 1000 per la scrittura e della linea 3500 per la lettura.

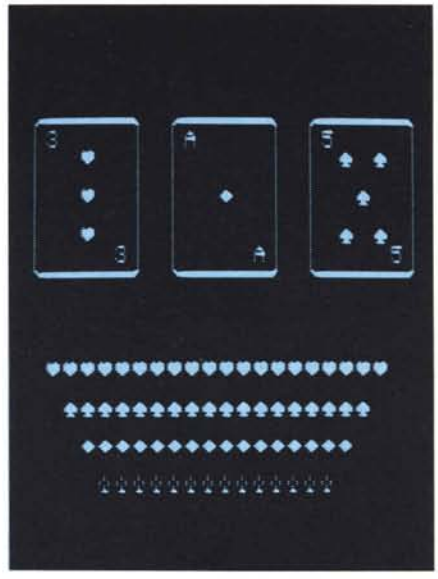
Il nome così ricavato viene introdotto come stringa B$ — premendo ora il tasto f7 — nella frase di apertura e scrittura del file contenuta nella linea 2000. Ogni singolo record di tale file è composto dal numero contenuto in locazioni video successive che, sequenzialmente, vengono memorizzate sul disco dal segmento contenuto tra le linee 2000 e 2050. Come potrete osservare guardando la linea 2020, ogni numero da trasferire viene trasformato in stringa, come è sempre utile fare, per avere meno problemi in lettura, nella gestione di file di questo tipo.

Per la lettura dello stesso file da disco, utilizzeremo i tasti f2 — per il nome — ed f8 — per il caricamento —. Il processo è analogo a quello descritto per il salvataggio quindi non è il caso di soffermarsi ancora su questi punti.

Video e tastiera come periferiche

Dalla linea 100 alla 200 viene simulato il





Esempio di schermate composte con il nostro editor.

lampeggio del cursore, ma il sistema da noi impiegato è forse il peggiore perché, se il cursore da noi prodotto incontra un carattere scritto in precedenza, lo cancella e siamo quindi costretti a riscriverlo. Questo fatto, in alcune situazioni che non stiamo ad elencare, crea notevoli difficoltà. Diciamo che la cosa è stata un po' voluta per mettere più in evidenza il metodo di input che vi illustriamo di seguito.

Abbiamo visto che, per mezzo di un file, ci possiamo mettere in comunicazione con un'unità periferica assegnando un opportuno numero ad uno dei parametri contenuti nella sintassi di apertura del file stesso. Il disco è identificato dal numero 8, la stampante dal numero 4 o 5 (6 per il plotter 1520), il registratore dal numero 1 e l'RS 232 dal numero 2.

Forse molti però non hanno mai pensato che le unità periferiche che usiamo più di frequente sono proprio il video e la tastiera. Esse, oltre ad essere impiegate nei consueti modi, possono essere gestite mediante un file. Facciamo qualche esempio.

Cominciamo dallo schermo. Il suo codice di periferica è 3 quindi, sfruttando la consueta sintassi di gestione, potremo scrivere:

```
OPEN 10,3
PRINT#3, "TOPO"
CLOSE 10
```

Questa serie di operazioni produrrà la scrittura sullo schermo della parola TOPO così come avverrebbe, ad esempio, su di una stampante se assegnassimo il numero 4 al posto del 3.

Allo stesso modo possiamo impiegare la tastiera, in maniera diversa da quella consueta, per prelevare dei dati dall'esterno e depositarli sul video. Anche qui la sintassi non lascia spazio alla fantasia: per il prelievo useremo un comando di Input# e l'indirizzo di periferica sarà 0.

Scriveremo cioè:

```
OPEN 5,0
INPUT#5,A$
CLOSE 5
```

e cominceremo ad introdurre i dati che saranno posti sullo schermo. Dopo il Run, avremo a disposizione il cursore che ci aiuterà ad orientarci sul teleschermo e apparentemente non noterete nessuna differenza: potrete cioè scrivere tutto ciò che vorrete come succedeva prima del Run, cioè l'editor di schermo è immutato. Ma attenzione! La macchina si trova in fase di elaborazione e, per rendervene conto, premete Return: il programma si arresterà e comparirà la scritta Ready.

```
10 REM ----------------------
20 REM ESEMPIO DI GESTIONE
30 REM DELLO SCHERMO VISTO
40 REM COME UNITA' PERIFERICA
50 REM ----------------------
70 POKE53280,0:POKE53281,0
80 PRINT"■":OPEN5,0
90 PRINT"□":POKE650,128
100 INPUT#5,A$
105 PRINT"■";
120 IFA$="Q"THENPRINT:GOSUB200
130 IFA$="W"THENPRINT:GOSUB300
140 IFA$="E"THENPRINT:GOSUB400
150 IFA$="R"THENPRINT:GOSUB500
160 GOTO100
161 REM ----------------------
200 PRINT"PRIMA DIRAMAZIONE"
210 RETURN
211 REM ----------------------
212 REM ----------------------
300 PRINT"SECONDA DIRAMAZIONE"
310 RETURN
311 REM ----------------------
312 REM ----------------------
400 PRINT"TERZA DIRAMAZIONE"
410 RETURN
411 REM ----------------------
412 REM ----------------------
500 PRINT"QUARTA DIRAMAZIONE"
510 RETURN
511 REM ----------------------
```

Figura 4 - Esempio di file di input da tastiera.

Il vantaggio dell'uso di un file del genere, per comporre le schermate nel nostro programmino dimostrativo della figura 1, è evidente: non dovrete più simulare nessun cursore e l'editor sarà perfetto. Le difficoltà cui accennavamo sono quindi superate.

Nella figura 4 riportiamo un listato che sfrutta le potenzialità descritte. Esso mostra come sia possibile scrivere sullo schermo e diramare verso uno dei quattro simbolici blocchi delle linee 200, 300, 400 e 500: basta digitare uno dei tasti Q, W, E o R avendo cura di premere Return portandosi su una linea vuota dello schermo. A questo punto non dovrebbe risultare troppo difficile modificare il nostro programma editore di schermate in maniera opportuna impiegando questa nuova tecnica.

Concludendo

Tutto quanto detto, con minime modifiche, può essere esteso al Vic 20. Esse dipendono per prima cosa dalla configurazione di memoria in cui esso si trova. Infatti, la memoria video di tale computer è... mobile spostando il suo punto di partenza dalla locazione 7680 alla 4096 a seconda che ci si trovi in configurazione base o in base + 3K oppure con più di 3K di Ram. La stessa cosa vale per il colore che passa da 38400 a 37888 e per la memoria riservata al programma.

Naturalmente la Ram video del Vic è la metà di quella del C 64 quindi andranno modificati opportunamente i cicli di For... Next. Altre modifiche vanno apportate alle linee 3 e 6 che, rispettivamente, selezionano l'area buffer (che dipenderà ora dalla configurazione in cui ci troviamo) e cambiano il colore dello schermo. Se ci avete seguiti attentamente fin'ora, effettuerete queste modifiche in pochi minuti. Se avrete difficoltà, scriveteci! MC