

# software

## C-64

a cura di Tommaso Pantuso

## Windows Operating System

di Tommaso Masi - Abano Terme

Il Windows Operating System (WOS) è un programma in linguaggio macchina che consente mediante chiamate ad alcune routine e in base a certi parametri, di poter impiegare 6 finestre sullo schermo di testo del Commodore 64.

È innanzitutto opportuno chiarire il concetto di «finestra» così com'è implementato dal WOS. Una finestra è una zona dello schermo delimitata da un riquadro, che la distingue e ne disegna il contorno; essa inoltre risulta completamente indipendente dallo schermo stesso. Ciò significa che quando una finestra viene aperta (cioè tracciata) si sovrappone a quanto è presente nella stessa zona dello schermo cancellandola, e tuttavia quando la finestra verrà chiusa (ovvero eliminata) al suo posto ricomparirà quanto era stato in precedenza cancellato. In pratica questo significa che quando una finestra viene chiusa è come se non fosse mai esistita: lo schermo tornerà all'aspetto precedente. Questo può fare intuire una prima ragione per cui risulta conveniente utilizzare le finestre come mezzo di interazione tra l'utente e il programma: ogni volta che devono comparire sullo schermo delle comunicazioni temporanee (directory, messaggi di aiuto o di errore), queste possono essere inquadrate in una finestra. I celebri menu «a tendina» del Macintosh, il primo computer che abbia adottato in modo sistematico questo sistema di output su video, sono delle finestre che occupano spazio per lo stretto tempo necessario affinché l'utente compia la sua scelta.

Ma una finestra non è solo un riquadro bensì un vero e proprio scher-

mo avente le stesse caratteristiche dello schermo di testo originale. Così quando la scrittura di una frase raggiunge il lato destro della finestra, essa continua nella riga sottostante a partire dal lato sinistro e, se non c'è più spazio, il contenuto della finestra scrollerà di una riga verso l'alto; se inviate in scrittura ad una finestra alcuni codici ASCII di controllo come il n. 147 (clear) o il n. 17 (cursor down) o il n. 20 (delete) otterrete gli effetti previsti ma non su tutto lo schermo, bensì solo all'interno della finestra che state usando. Inoltre in fase di input, l'editor di schermo del Commodore 64 viene adattato alle dimensioni della finestra. Ecco perché il WOS permette tecniche di output su video del tutto nuove: in sostanza il programmatore dispone di ben 6 schermi di testo del tutto indipendenti tra loro, dimensionabili e posizionabili secondo le esigenze del programma. Questo permette, anche grazie all'uso del colore e dell'effetto «reverse», di ottenere visualizzazioni di dati estremamente comprensibili e molto più efficaci di quelle standard.

Si impongono a questo punto delle note che specificano le modalità con cui è possibile operare su più finestre «contemporaneamente». In effetti in ogni istante si può scrivere, leggere in input e compiere le altre operazioni consentite dal WOS, solo su una finestra: quella che in quel momento risulta abilitata all'uso. La finestra abilitata risulta sovrapposta a tutte le altre ed appare in «primo piano» agli occhi dell'utente. Per chiarire il concetto è utile rifarsi alla metafora della scrivania tipica del Macintosh e di molti computer e S.O. dell'ultima generazione (Amiga, Atari ST, GEM, MSWindows). Immaginate perciò che lo schermo del vostro C64 rappresenti la superficie di un tavolo: se volete scrivervi sopra, la prima cosa da fare è procurarsi un foglio: aprite una finestra. Se nel corso del lavoro è necessario leggere altri dati o compiere operazioni ausiliarie è più ordinato, e più comprensibile, servirsi di altri fogli invece di riempire quello su cui state la-

vorando: quando vedete uno schermo pieno di finestre sovrapposte e mezze nascoste, dovete pensare ad una scrivania ingombra di fogli. Se nel corso del lavoro il vostro primo foglio è finito sotto agli altri come fate per scrivervi sopra? Semplice, lo afferrate per quell'angolo che spunta sotto agli altri e lo ponete sopra a tutti: la stessa cosa avviene quando abilitate una finestra del WOS: la estraete dal mucchio in modo da poterci lavorare.

Una volta abilitata una finestra, e ciò avviene in modo automatico all'apertura, è possibile cancellarla, spostarla sullo schermo, scrivervi dentro e leggersi in input, eventualmente posizionando il cursore al suo interno con coordinate relative, e persino salvarla su un dischetto. Quest'ultima possibilità offerta dal WOS può semplificare in alcuni casi il lavoro al programmatore: quando occorrono più di 6 finestre in quanto nessuna può essere persa, cioè chiusa, senza danno per il pro-

### Per chi vuole il listato

Il listato di questo programma è lungo alcuni metri. In conseguenza di ciò, si è ritenuto opportuno non pubblicarlo, sia perché avrebbe occupato troppo spazio sulla rivista sottraendone ad altri argomenti, sia perché una digitazione senza errori di un listato così lungo appare poco probabile.

Chi è interessato al programma può ordinare, secondo il solito sistema, il disco o la cassetta in redazione. È anche possibile «pescare» direttamente (e gratuitamente) il programma per via telematica, dal nostro servizio MC-Link; questo ovviamente vale per chi è attrezzato in tal senso. Ricordiamo che per ottenere una casella su MC-Link è sufficiente telefonare (con un modem e un programma di comunicazione) al numero 06/4510211.

Questo programma è disponibile su disco presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 185.

gramma, è possibile salvarne una su disco, cancellarla dallo schermo e avere così l'opportunità di crearne un'altra. Quando sarà necessario riutilizzarla, la si potrà caricare da disco e servirsi come se non fosse mai stata cancellata. Inoltre in quei programmi che prevedono una funzione di help risulta molto pratico memorizzare su disco le schermate di informazioni sotto forma di finestre, ottenendo anche un grande risparmio di memoria centrale. Questa funzione è stata prevista solo per l'uso con un disk-drive (avente numero di device 8) in quanto l'utilizzazione con cassette sarebbe stata particolarmente macchinosa non essendo possibile su questo dispositivo effettuare degli accessi di tipo random.

Adesso saranno esaminate in dettaglio tutte le operazioni del WOS al fine di creare e utilizzare finestre.

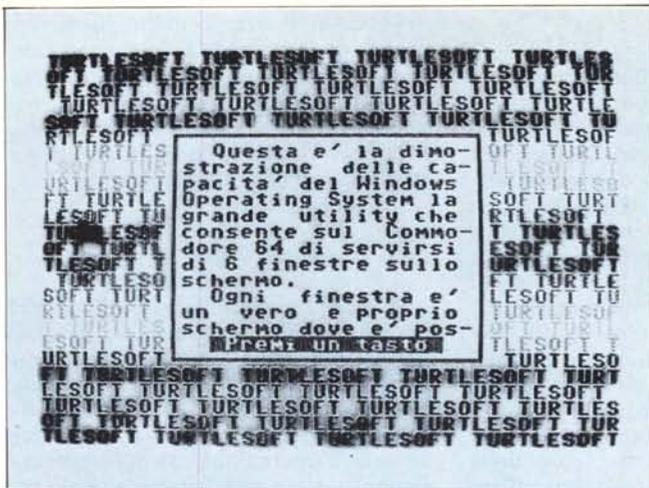
### Definizione di una nuova finestra

L'apertura di una nuova finestra necessita di 7 parametri che devono essere comunicati in una specifica zona di memoria del WOS. Questi sono la colonna e la riga dell'angolo in alto a sinistra della finestra (VX) e (VY), la lunghezza orizzontale e verticale (LO) e (LV), il colore della finestra (CC), il suo numero (NW) ed infine l'indicatore del modo di visualizzazione (RV).

Le coordinate dell'angolo in alto a sinistra, che da ora in poi sarà chiamato vertice della finestra, e le lunghezze dei lati, determinano la posizione e la dimensione della finestra. Questi dati devono essere interpretati esattamente per non incorrere in errori. VX può variare da 0 a 38 e VY da 0 a 23, essendo i loro valori misurati nel modo convenzionale adottato dalla Commodore per lo schermo di testo: la coppia (0,0) rappresenta l'angolo in alto a sinistra dello schermo e (39,24) quello opposto. LO può variare da 2 a 40 e LV da 2 a 25: questi valori misurano i lati della cornice che, essendo costituita almeno da 4 angoli, non può essere più piccola di un quadrato 2x2 (anche se in questo caso è impossibile scrivervi dentro qualcosa all'interno).

In base a queste considerazioni dovrebbe essere chiaro che una finestra avente come parametri VX=0, VY=0, LO=40 e LV=25 è la più grande possibile in quanto occupa tutto lo schermo e al suo interno, data la presenza della cornice, si possono scrivere 23 righe di 38 caratteri ciascuna.

Nella determinazione di questi parametri bisogna evitare valori di LO e di LV che, data la posizione del vertice, facciano fuoriuscire dallo schermo parte della finestra. In tal caso il WOS si accorgerà dell'errore e si rifiuterà di aprire la finestra.



CC può assumere valori da 0 a 15 in conformità ai codici dei colori del Commodore 64 e rappresenta il colore con il quale verrà aperta la finestra, ovvero il colore della cornice e dei caratteri stampati al suo interno almeno finché tra questi non vi sarà un codice ASCII di controllo del colore di scrittura.

Il valore NW può variare da 1 a 6 e rappresenta il numero con il quale il WOS identifica la finestra stessa. In merito a questo parametro è necessario fare una considerazione molto importante ai fini di un uso corretto del WOS. Definire una finestra con un numero NW già utilizzato per un'altra è possibile ma assolutamente sconsigliabile a chi non abbia raggiunto una certa dimestichezza con il WOS stesso. Questo procedimento può essere causa di «strani» risultati in determinati casi che sarebbe troppo complicato descrivere. Se volete approfondire questo problema provate a compiere i tre seguenti esperimenti:

- aprite una finestra, poi una seconda con un valore NW diverso e che NON si sovrapponga alla prima; quindi aprite una terza finestra con lo stesso numero NW della seconda: muovete le due finestre gestibili dal WOS (la seconda non lo è più a causa dell'apertura della terza con lo stesso valore NW) e osservate i risultati;

- aprite una finestra, poi una seconda con un valore NW diverso e che si SOVRAPPONGA in parte alla prima; quindi aprite una terza finestra con lo stesso numero NW della seconda: muovete le due finestre gestibili dal WOS e osservate i risultati;

- aprite due finestre con due valori NW diversi, poi una terza con un altro valore di NW e che si SOVRAPPONGA ad entrambe; quindi aprite una quarta finestra con lo stesso numero NW della terza: muovete le tre finestre gestibili dal WOS e osservate i risultati.

Nel primo esperimento tutto funzio-

nerà nel modo corretto, negli altri due potrete osservare gli «strani» risultati di cui sopra.

In ogni caso per non incorrere in problemi è sufficiente utilizzare solo 6 finestre attribuendo ad ognuna un numero di riconoscimento NW diverso.

Concludiamo la descrizione dei parametri necessari alla definizione di una nuova finestra con alcune notizie sul parametro RV. Questo indica se la finestra deve essere visualizzata in modo normale (RV=0) o in «reverse» (RV=-1). Nel primo caso comparirà una cornice di colore CC con all'interno uno spazio vuoto, in quanto la finestra viene automaticamente pulita all'apertura. Nel secondo apparirà tutta la finestra di colore CC, mentre la cornice avrà il colore dello sfondo dello schermo. Si noti che se RV=1 e non viene inviato in scrittura il codice ASCII n. 146 (reverse off) tutti i caratteri stampati all'interno della finestra, anche quelli immessi da tastiera durante un input, compariranno in «reverse». Inoltre il codice ASCII n. 147 (clear) riempirà la finestra di spazi in reverse (in effetti pulire lo schermo significa riempirlo di spazi) e l'effetto finale sarà quello di vedere un'area di schermo pulita colorata in base al colore in uso, che per default è CC. Tuttavia come è possibile scrivere in «reverse» in una finestra normale inviando prima il codice ASCII n. 18 (reverse on), così si può scrivere normalmente in una finestra in «reverse» inviando prima il codice ASCII n. 146.

Questa caratteristica del WOS permette di usare le finestre dei due tipi semplicemente agendo sul parametro RV. È immediato notare che su uno schermo normale una finestra in «reverse» risalta nettamente agli occhi dell'utente e pertanto è particolarmente adatta per indicare condizioni di errore o di emergenza del programma.

Al vantaggio offerto da questa caratteristica delle finestre si contrappone un piccolo problema che può esse-

re risolto con un semplice accorgimento: la routine di stampa del Commodore 64 fa sì che di seguito ad un codice ASCII n. 13 (return) corrisponda un ritorno automatico del modo «reverse off». Questo porta all'abitudine di scrivere parole in «reverse» senza curarsi di inviare al termine il codice ASCII di controllo n. 146 in quanto il «return», che è addirittura automatico nel Basic se lo statement PRINT non termina con un punto e virgola, mette lo schermo in modo normale. Questo NON avviene con la routine di stampa del WOS, pertanto quando in una finestra normale stampate delle parole in «reverse» ricordatevi di inviare al termine il codice ASCII n. 146 per tornare a scrivere normalmente.

Per concludere, sull'apertura di una finestra resta da dire che il cursore verrà posizionato nell'angolo in alto a sinistra della stessa nella zona utile, perciò non sul vertice della cornice, ma sulla posizione di coordinate  $VX + 1$  e  $VY + 1$ . Questa è anche la posizione su cui verrà posto il cursore in seguito alla stampa dei codici ASCII di controllo n. 147 (clear) e n. 19 (home).

### Caricamento da disco di una finestra

È il secondo mezzo concesso dal WOS per aprire una finestra: i parametri richiesti sono il nome del file con il quale la finestra è memorizzata su disco, che va posizionato in una zona di memoria detta buffer di input/output del WOS con un byte nullo al termine, e il valore NW in base al quale identificarla successivamente. Tutti gli altri parametri sono infatti memorizzati insieme al contenuto della finestra sul dischetto. In effetti dal punto di vista pratico il caricamento di una finestra pulita dà lo stesso risultato della definizione di una finestra nuova.

Sono ammesse due tecniche di caricamento: una visualizza la finestra là dove questa si trovava al momento del salvataggio, l'altra permette di ottenerla in una posizione diversa semplicemente indicando i nuovi valori di VX e di VY. Ovviamente nell'indicarli è necessario attenersi alle precauzioni viste sopra in merito, per non incorrere in errori.

In entrambi i casi il caricamento ripristina la stessa situazione presente al momento del salvataggio posizione del cursore all'interno della finestra, colore di scrittura dei caratteri e modo di visualizzazione (normale o in «reverse»). Richiamando la metafora della scrivania è come se avessimo ripreso un foglio da un cassetto.

In caso di errore nelle operazioni con il dischetto (disk-drive non connesso, file non presente nella directory) il caricamento è abortito senza che il programma venga bloccato. In una locazione di memoria compare un valore che indica il tipo di errore:

- 0 - tutto OK
- 2 - file già aperto
- 4 - file non trovato
- 5 - disk-drive non presente

Al fine di evitare l'errore di codice 2 in programmi che accedono al disk-drive si consiglia di non utilizzare i numeri logici di file 72 e 73 (usati dal WOS) e i canali di comunicazione 14 e 15. Questi canali vengono chiusi al termine dell'operazione con il dischetto e pertanto non sono disponibili solo per il tempo necessario al WOS per operare.

### Abilitazione di una finestra

È indispensabile effettuare questa operazione per poter operare su una finestra e, come visto sopra, viene eseguita automaticamente al momento dell'apertura o del caricamento da disco. È necessario comunicare il parametro NW (con un valore tra 1 e 6) prima di chiamare l'apposita routine. La finestra abilitata viene posta in «primo piano» eventualmente sovrapponendosi ad altre. Il cursore verrà posto all'interno della finestra nella stessa posizione dove si trovava l'ultima volta che la finestra è stata utilizzata, il colore di scrittura dei caratteri e il modo di visualizzazione saranno quelli che erano abilitati nel momento in cui la finestra è stata abbandonata. Questi particolari devono essere tenuti ben presenti per non ottenere dei risultati indesiderati.

Se, per sbaglio, viene comunicato il numero NW della finestra già abilitata o un valore non ancora attribuito a qualche finestra il programma non si bloccherà in quanto queste situazioni vengono trattate dal WOS senza ricorrere a procedure d'errore.

### Scrittura in una finestra

Una volta abilitata in uno dei tre modi visti sopra, la finestra è pronta per essere utilizzata. Ciò significa anzitutto poterci scrivere all'interno. La stringa che si intende stampare deve essere memorizzata nel buffer di I/O del WOS con un byte nullo al termine che indichi alla routine di scrittura del WOS dove termina la stringa. Qualora si intenda formattare la stringa in modo da non troncare le parole in fondo alle varie righe è opportuno ricordare che queste possono contenere LO-2 caratteri.

Alcuni codici ASCII, inutili in fase di stampa, non sono accettati e ignora-

ti quando vengono incontrati: questi sono il n. 20 (delete), il n. 148 (insert) e i codici corrispondenti ai tasti funzione (dal n. 133 al n. 140). Tutti gli altri codici ASCII di controllo funzionano regolarmente benché in riferimento alla finestra in uso; per esempio, i codici di movimento del cursore rispettano le dimensioni della finestra. Tra questi è interessante notare che il codice n. 145 (cursor up) se il cursore si trova nella prima linea della finestra, e il codice n. 157 (cursor left) se il cursore si trova nell'angolo in alto a sinistra, provocano uno scrolling di una riga verso il basso del contenuto della finestra, operando perciò in modo opposto a quanto avviene anche sullo schermo quando viene raggiunta l'ultima linea. In altre parole, mentre sullo schermo lo scrolling è possibile solo in una direzione, nelle finestre è possibile in entrambe.

Tutte le altre caratteristiche della stampa del WOS sono assolutamente analoghe a quelle che il S.O. del C64 permette sullo schermo. C'è tuttavia da notare che la routine del WOS agisce direttamente sullo schermo anche se è stata modificata la periferica di output di default (come per esempio con lo statement CMD del BASIC).

### Input all'interno di una finestra

L'altra funzione basilare che deve poter essere svolta sullo schermo è quella di input. All'interno di una finestra non può essere utilizzato il normale input del S.O. del C64 in quanto l'editor utilizzato è «a pieno schermo». Inoltre con l'input tradizionale se per errore premete i tasti SHIFT e CLR/HOME otterrete la pulizia dello schermo danneggiando l'aspetto del programma e, in molti casi, anche la funzionalità dell'immissione; inoltre potete modificare il colore in uso per la scrittura e abilitare il modo «reverse» con le consuete procedure da tastiera: queste caratteristiche, se sono utili in fase di scrittura di un programma durante la sua battitura, sono deleterie durante l'esecuzione di un programma che chiede dei dati in input in quanto ne rovinano l'aspetto esteriore.

L'input del WOS è di tipo controllato. Ciò significa che:

- bisogna specificare il numero massimo di caratteri accettati (da 1 a 255);

- alcune funzioni di edit sono disabilitate.

L'editor infatti non è più «a pieno schermo» ma è «di linea»: i tasti utilizzabili per le correzioni sono solo l'INST/DEL e il CRSR orizzontale, SHIFTati o meno.

L'immissione termina alla pressione del tasto RETURN: il WOS legge la stringa dallo schermo e la posiziona

nel suo buffer di I/O con un byte nullo al termine. Si tenga presente che non sono previsti caratteri separatori e questo permette di utilizzare virgole, punti e virgola e due punti senza rischiare di perdere parti della stringa immessa. Il WOS non distingue tra input numerici ed alfabetici: nel buffer di I/O viene posta la stringa così come appare sullo schermo tranne gli eventuali spazi posizionati dopo l'ultimo carattere della stringa stessa al momento della pressione del tasto RETURN. Naturalmente durante l'immissione la posizione del cursore è indicata dalla locazione di schermo lampeggiante.

### Posizionamento del cursore

È una funzione del WOS che, utilizzata accanto alla scrittura e all'input permette di avere il pieno controllo dell'aspetto del testo all'interno delle finestre. È necessario comunicare le coordinate della locazione in cui si desidera il cursore ma non con i consueti valori assoluti ( $X=0-39$ ,  $Y=0-24$ ) bensì con misure relative. La coppia  $X=1$  e  $Y=1$  indica l'angolo in alto a sinistra; con  $X=LO-2$  e  $Y=LV-2$  si posizionerà il cursore nell'angolo opposto; con valori intermedi di  $X$  (numero di colonna) e  $Y$  (numero di riga) è possibile raggiungere ogni locazione della finestra. Valori maggiori o minori di questi estremi non vengono accettati: la routine si interrompe senza posizionare il cursore.

### Spostamento di una finestra

Con questa routine del WOS è possibile modificare i valori dei parametri  $VX$  e  $VY$ : comunicati i nuovi valori e chiamata la routine si vedrà la finestra portarsi istantaneamente nella nuova posizione. Ovviamente è necessario, nell'indicare questi nuovi valori, attenersi alle indicazioni viste nel caso dell'apertura della finestra per evitare di portare la finestra, anche in parte, al di fuori dello schermo.

### Salvataggio su disco di una finestra

Unico dato richiesto è il nome del file con il quale si vuole salvare la finestra comunicandolo nel buffer di I/O con un byte nullo al termine: il nome non deve essere più lungo di 16 caratteri come richiesto dal DOS del C64: se la stringa è più lunga, quelli eccedenti verranno scartati. Il file ottenuto su disco è di tipo sequenziale.

Per gli errori con il disk-drive si veda il paragrafo relativo al caricamento.

La finestra salvata non viene cancellata dallo schermo e rimane abilitata.

### Cancellazione di una finestra

Questa routine, che non necessita di parametri in quanto opera sulla finestra abilitata, elimina la finestra dallo schermo sostituendola con lo sfondo corrispondente, e azzerà i suoi parametri rendendo disponibile il suo numero  $NW$  per un'altra finestra. L'operazione non è reversibile e pertanto va attuata con prudenza in quanto la finestra viene definitivamente persa.

Se quella cancellata non è l'unica presente sullo schermo, viene automaticamente abilitata la finestra che era stata usata in precedenza. In effetti le varie finestre sono viste in base ad un grado di priorità attribuito dinamicamente in base alle abilitazioni: se si aprono in sequenza tre finestre rispettivamente con i numeri 1, 3 e 4 a quest'ultima verrà attribuito il grado di priorità maggiore; se successivamente viene abilitata la numero 1 questa otterrà il primo grado di priorità facendo scivolare al secondo la 4 e al terzo la 3. Cancellando la finestra 1 verrà perciò automaticamente abilitata la 4.

Passiamo ora a descrivere la struttura in memoria del WOS e delle sue routine. Il codice in linguaggio macchina è posizionato dalla locazione 49152-\$C000 a 52207-\$CBEF. Di seguito per 256 byte si trova il buffer di I/O da 52208-\$CBF0) a 52463-\$CCEF. Il WOS utilizza 8 locazioni in pagina 3 da 820-\$0334 a 827-\$033B. Inoltre per conservare tutti i dati necessari a maneggiare contemporaneamente le finestre è necessaria una notevole quantità di memoria: a questo scopo il WOS utilizza la RAM solitamente nascosta dalla ROM del Basic e del S.O.: si tratta di 16 Kbyte divisi in due blocchi uguali e posizionati uno da 40960-\$A000 a 49151-\$BFFF e l'altro da 57344-\$E000 a 65535-\$FFFF.

Ciò significa che viene lasciata libera per il programmatore tutta l'area usualmente destinata al codice Basic, più 784 byte posti da 52464-\$CCF0 a 53247-\$CFFF. Per quanto riguarda le locazioni di pagina zero, restano disponibili le solite da 251-\$00FB a 254-\$00FE.

In conclusione segue la lista degli indirizzi di chiamata e dei parametri delle singole routine del WOS.

### Definizione di una nuova finestra

Indirizzo di inizio: 51603-\$C993 Parametri:

$VX \rightarrow 50030$ -\$C36E  
 $VY \rightarrow 50031$ -\$C36F  
 $LO \rightarrow 50032$ -\$C370  
 $LV \rightarrow 50033$ -\$C371  
 $CC \rightarrow 50034$ -\$C372  
 $NW \rightarrow 50035$ -\$C373  
 $RV \rightarrow 50036$ -\$C374

### Caricamento semplice da disco

Indirizzo di inizio: 52171-\$CBCB Parametri:

$NV \rightarrow 50035$ -\$C373  
 nome del file nel buffer di I/O con byte nullo alla fine

Note: nella locazione 52207-\$CBEF compare il codice d'errore delle operazioni con il disk-drive.

### Caricamento da disco con spostamento

Indirizzo di inizio: 52174-\$CBCE Parametri:

$VX \rightarrow 50030$ -\$C36E  
 $VY \rightarrow 50031$ -\$C36F  
 $NW \rightarrow 50035$ -\$C373  
 nome del file nel buffer di I/O con byte nullo alla fine

Note: nella locazione 52207-\$CBEF compare il codice d'errore delle operazioni con il disk-drive.

### Abilitazione di una finestra esistente

Indirizzo di inizio: 51630-\$C9AE Parametri:

$NW \rightarrow 50035$ -\$C373

### Scrittura di una finestra

Indirizzo di inizio: 49411-\$C103 Parametri:

la stringa nel buffer di I/O con byte nullo alla fine

### Input dentro una finestra

Indirizzo di inizio: 51639-\$C9B7 Parametri:

il numero massimo di caratteri di 50038-\$C376

Note: la stringa immessa viene posta nel buffer di I/O con un byte nullo al termine.

### Posizionamento del cursore

Indirizzo di inizio: 49807-\$C28F Parametri:

$X \rightarrow 50036$ -\$C374  
 $Y \rightarrow 50037$ -\$C375

### Spostamento di una finestra

Indirizzo di inizio: 50039-\$C377 Parametri:

$VX \rightarrow 50030$ -\$C36E  
 $VY \rightarrow 50031$ -\$C36F

### Salvataggio su disco

Indirizzo di inizio: 51808-\$CA60 Parametri:

nome del file nel buffer di I/O con byte nullo alla fine

Note: nella locazione 52207-\$CBEF compare il codice d'errore delle operazioni con il disk-drive.

### Cancellazione di una finestra

Indirizzo di inizio: 51612-\$C99C

Note: viene abilitata la finestra precedente se esiste.

Terminiamo la parte relativa al WOS con due indicazioni di uso pratico. La prima riguarda il caricamento in memoria del codice in linguaggio macchina:

```
in modo diretto battere da tastiera
LOAD «WOS 1.3», 8,1 (RETURN)
e di seguito
NEW (RETURN)
```

oppure inserire come prime istruzioni del programma che utilizza il WOS le istruzioni BASIC, o il loro equivalente in altri linguaggi, seguenti

```
0 IF A=0 THEN A=1: LOAD«WOS 1.3»,
8,1
```

La seconda indicazione riguarda l'inizializzazione del WOS. Questa, necessaria al fine di un corretto funzionamento, deve essere inserita all'inizio del vostro programma e consiste nell'azzeramento di alcune locazioni di memoria: più precisamente dalla 50030-\$C36E alla 50038-\$C376 e dalla 50992-\$C730 alla 50999-\$C737.

In assembler si tratta di eseguire il seguente segmento:

```
RESET      LDA # $00
           LDY # $09
LOOP1      STASC36D,Y
           DEY
           BNE LOOP1
           LDY # $08
LOOP2      STASC72F,Y
           DEY
           BNE LOOP2
```

Mentre operando in BASIC si può scrivere:

```
1 FOR I=50030 TO 50038: POKE I,0:
NEXT I
2 FOR I=50992 TO 50999: POKE I,0:
NEXT I
```

Questa procedura resetta i registri del WOS facendogli perdere la possibilità di operare con le finestre utilizzate in precedenza e può anche essere usata quando, per pulire completamente uno schermo contenente molte finestre, si preferisca inviare al video il codice ASCII n. 147 (clear) invece di cancellare ad una ad una ogni finestra.

## WBASIC

Il WBasic è un'estensione del Basic V2 del Commodore 64 costituita da 8 nuove istruzioni che consentono di accedere facilmente a tutte le caratteristiche del WOS. Tuttavia il WBasic costituisce un programma a sé in quanto, per problemi di occupazione di memoria, contiene una versione del WOS leggermente diversa dalla 1.3 che non è utilizzabile separatamente dalle istruzioni del WBasic.

Per caricare in memoria il WBasic battere da tastiera:

```
LOAD«WBasic 1.0», 8,1 (RETURN)
```

e di seguito:

```
NEW (RETURN)
infine per collegarlo all'interprete residente battere:
```

```
POKE 782,6: SYS 49835: POKE 56, 159:
POKE 644, 159 (RETURN)
```

oppure, se si desidera il suo caricamento automatico dal vostro programma, usare all'inizio le linee seguenti:

```
0 IF A=0 THEN A=1: LOAD«WBASIC
1.0», 8,1
```

```
1 POKE 782,6: SYS 49835: POKE 56, 159:
POKE 644, 159: CLR
```

Una volta abilitato, il WBasic non modifica i colori del video, né scrive sullo schermo un messaggio di presentazione: pertanto è l'ideale per essere utilizzato nei vostri programmi.

Il WBasic occupa tutti i 4 Kbyte posti da 49152-\$C000 a 53247-\$CFFF, inoltre si serve degli ultimi 256 byte posti da 40704-\$9F00 a 40959-\$9FFF, ovvero al termine dell'area di memoria riservata ai programmi in Basic per i quali restano perciò disponibili 38655 byte. Inoltre utilizza le altre zone di memoria necessarie al WOS.

Valgono anche per il WBasic le indicazioni viste a proposito dell'inizializzazione del WOS: perciò è opportuno che tra le prime righe del vostro programma vi siano le seguenti:

```
2 FOR I=50030 TO 50038: POKE I,0:
NEXT I
3 FOR I=50992 TO 50999: POKE I,0:
NEXT I
```

Per quanto riguarda il funzionamento dell'interprete, le nuove istruzioni sono pienamente compatibili con quelle residenti; benché solo per l'istruzione di input sia previsto il messaggio d'errore ILLEGAL DIRECT, è superfluo rilevare che le istruzioni del WBasic sono utili solo se utilizzate all'interno di un programma.

Un'altra considerazione va fatta in merito ai comandi vuoti che non sono ammessi dal WBasic. Perciò una linea di programma come:

```
20 POKE 198,0::GET A$
```

(si notino i due punti consecutivi) darà luogo a un:

```
? SYNTAX ERROR IN 20
```

Una limitazione più seria è invece l'impossibilità di inserire un'istruzione del WBasic di seguito al THEN di un IF. Ad esempio la linea:

```
100 IF A>0 THEN WINDOW (1)
```

non è lecita. Tuttavia un piccolo stratagemma permette di risolvere il problema: infatti se battete la linea nel seguente modo:

```
100 IF A>0 THEN WINDOW(1)
```

tutto funzionerà perfettamente. Purtroppo il poco spazio lasciato libero dal WOS, e la decisione di ridurre al minimo l'area di memoria destinata al programma in Basic, non ha permesso di eliminare questi inconvenienti che comunque non limitano assolutamen-

te né la possibilità del programmatore, né le prestazioni del WBasic.

Passiamo ora alla descrizione della sintassi e del funzionamento delle singole istruzioni tenendo presente che le modalità d'uso sono quelle già viste a proposito delle corrispondenti routine del WOS e che tutti i parametri delle nuove istruzioni possono essere costituiti da espressioni numeriche o di tipo stringa esattamente come per le istruzioni residenti.

### WINDOW

Sintassi

WINDOW (nw) oppure

WINDOW (nw) vx, vy, lo, lv, cc, rv

Nella prima forma abilita la finestra numero nw, nella seconda forma definisce una finestra nuova in base ai parametri passati. Per quanto riguarda i valori dei parametri si rimanda all'apposito paragrafo delle istruzioni del WOS. In caso di errore il programma si interrompe con il messaggio d'errore ILLEGAL QUANTITY.

### WLOAD

Sintassi

WLOAD (nw) «nome del file» oppure

WLOAD (nw) «nome del file», vx, vy

Nella prima forma carica la finestra indicata e la abilita con il numero nw, nella seconda forma la carica posizionandola sullo schermo in base ai valori vx e vy indicati.

### WRITE

Sintassi

WRITE espressione da stampare

Scriva nella finestra l'espressione. Quest'ultima segue le stesse regole del classico statement PRINT, eccetto per la presenza delle funzioni TAB, SPC, e del separatore virgola, che non sono accettati. Si noti che nel seguente contesto:

```
100 OPEN 4,4: CMD 4
```

```
110 PRINT «ORA SCRIVO SU CARTA»
```

```
120 WRITE «MENTRE ADESSO IN UNA
FINESTRA»
```

l'istruzione WRITE è corretta e funziona perfettamente.

### WINPUT

Sintassi

WINPUT v, lm oppure

WINPUT «messaggio»; v, lm

Riceve in input al massimo lm caratteri (con lm compreso tra 1 e 87) e li immette nella variabile v. La sintassi rispetta le regole del normale statement INPUT che accetta un messaggio da stampare solo se contenuto tra virgolette. Tuttavia non è possibile come nell'INPUT tradizionale leggere

più variabili contemporaneamente con la forma INPUT a,b,c. Inoltre non compare il classico punto interrogativo.

### WLOCATE

**Sintassi**  
WLOCATE x,y  
Posiziona il cursore all'interno della finestra con coordinate relative. Si faccia attenzione alle precauzioni riportate nell'apposito paragrafo delle istruzioni del WOS perché in caso di errore questa istruzione abortisce senza emettere messaggi d'errore, cioè senza interrompere il programma BASIC.

te nell'apposito paragrafo delle istruzioni del WOS perché in caso di errore questa istruzione abortisce senza emettere messaggi d'errore, cioè senza interrompere il programma BASIC.

### WMOVE

**Sintassi**  
WMOVE vx, vy  
Sposta la finestra in base alle nuove coordinate del vertice.

### WSAVE

**Sintassi**  
WSAVE «nome del file»  
Salva su disco la finestra abilitata.

### WERASE

**Sintassi**  
WERASE  
Cancella la finestra abilitata.

## Autostart Generator

di Luigi Tavolato - Roma

L'Autostart Generator crea il loader per un programma scritto in Basic o anche in Linguaggio Macchina (o compilato) a patto che inizi nella area standard riservata al Basic e sia preceduto dalla SYS corrispondente. Il loader andrà automaticamente in esecuzione non appena caricato con il ,8,1 (o con il ,1,1).

Il disassemblato della routine di Autostart è:

```
01F5-0202      '02' start vector
0203 LDA #009
0205 STA $D011 Disabilita il Video Chip (Blanking ON)
0208 LDA #100   Flag Per disattivare i messaggi del Kernel Per la SETMSG
0209 JSR $FF90 SETMSG: routine Kernel Per il controllo dei messaggi
020D LDA #002   Numero del File Logico Per la routine di SETLFS
020F LDA #002   Numero del dispositivo corrente (08 drive=01 registratore)
0211 LDV #001   Indirizzo secondario
0213 JSR $FF8A SETLFS: Predisporre il file logico
0216 LDA #nm    Lunghezza del nome del nome del file da caricare
0218 LDV #244   Low byte della Posizione in memoria del nome del file
021A LDV #002   high byte (#0244)
021C JSR $FF8D SETNAM: Predisporre il nome del file da caricare
021F LDA #000   Imposta l'accumulatore a caricamento
0221 LDV #000   Sia il registro X che quello Y vengono impostati a 0
0223 LDV #000   non dovendo essere effettuato un caricamento rilocato
0225 JSR $FFD5 LOAD: carica dal drive il programma nella RAM
0228 STX #2D    Low byte di fine Programma
022A STX #2E    High byte di fine Programma
022C LDA #000   Low byte del puntatore al byte corrente del testo BASIC
022E STA #7A
0230 LDA #008   High byte
0232 STA #7B
0234 JSR $A650   Esegue il comando "CLR"
0237 LDA #000
0239 STA $D011 Riabilita il Video Chip (Blanking OFF)
023C LDA #0EA
023E STA $0328 Disabilita i tasti RUN-STOP/RESTORE ed il LIST
0241 JMP #A7RE   RUN
0244-0253      nome del Programma
```

La prima parte del loader è costituita da una sequenza di 02 che vanno a sovrasciversi nell'area che il microprocessore usa per la gestione dello stack. In questo stack il 6502 memorizza, quando viene eseguita una JSR (l'equivalente del GOSUB del Basic), l'indirizzo di ritorno a cui salterà non appena incontri, nella subroutine chiamata, un'istruzione di RTS (Return From Subroutine). La sequenza di 02 provocherà l'esecuzione della routine in L.M. che inizia dalla locazione \$0203. Quest'ultima caricherà e manderà in esecuzione il programma una volta predisposti gli opportuni parametri.

La disabilitazione del video-chip (blanking), riducendo il numero di operazioni che l'Interrupt del sistema operativo deve compiere ogni qualvolta ne viene richiesta l'esecuzione (60 al secondo), ha lo scopo di accelerare l'operazione di caricamento del programma.

**SBA (186)** è la locazione di memoria in cui viene immesso il numero della periferica su cui stiamo operando (viene valorizzata non appena digitiamo LOAD «loader»,8,1 o LOAD «loader»,1,1 e premiamo RETURN). È usata per sapere se il programma deve essere caricato da nastro o da

disco, in modo che il loader possa funzionare correttamente su entrambi.

**\$A7AE (52926)** è la routine dell'interprete Basic che effettua l'esecuzione del successivo statement del programma in memoria. L'indirizzo dell'istruzione da eseguire è contenuto nelle locazioni \$7A e \$7B (122-123), nella consueta forma byte basso-byte alto.

Il loader si distrugge, anche se solo in parte, non appena viene eseguito il RUN, poiché occupa il buffer di INPUT del sistema, un'area RAM frequentemente usata dall'interprete e che si estende dalla locazione \$0200 alla \$0258 (512-600).

Se usate il loader come sistema di protezione, considerate che, se il programma da proteggere viene caricato indipendentemente, il sistema non può funzionare. È opportuno allora nascondere nel programma dei test che controllino se è stato effettivamente caricato dal loader o no, andando a vedere, ad esempio, se la locazione 808 contiene il valore 234, o se il nome del programma caricato è proprio quello previsto (locazioni 580-596), o se sono presenti in memoria gli ultimi byte del loader (che non dovrebbero cancellarsi immediatamente). Se viene riscontrata qualche imprecisione, non è un grosso problema far sì che il programma non funzioni a dovere.

Se il loader viene usato su nastro, il programma da caricare deve essere salvato subito di seguito ad esso.

```
100 REM*****
110 REM*
120 REM* GENERATORE DI AUTOSTART *
130 REM*
140 REM* PER PROGRAMMI IN BASIC *
150 REM*
160 REM*
180 REM*
190 REM***
200 REM*** LUIGI TAVOLATO ***
210 REM***
220 REM*
230 REM*
240 REM*
250 REM*
260 REM*
280 REM*****
290
1000 DATA 169,0,32,144,255,173,17,208,41,239,153,17,208
1005 DATA 169,2,166,186,160,1,32,186,255
1010 DATA 169,256,162,78,160,2,32,189,255,162,255,160,255,169,0,32,213,255
1020 DATA 134,45,132,46,169,0,133,122,169,8,133,123,32,96,166
1030 DATA 169,225,153,40,3,173,17,208,9,14,153,17,208,76,174,167,0
1050 POKE$3280,0:POKE$3281,0
1100 PRINT:(CLR)
1101 PRINT:(RVS) AUTOSTART GENERATOR (OFF)";
1102 PRINT:
1105 PRINT:(DOWN) BY LUIGI TAVOLATO
1106 PRINT:(DOWN) TEL. 06-636623
1120 INPUT:(DOWN) NOME AUTOSTART " :AS:1-LEN(AS):IFL>16ORL=OTHENRUN
1130 INPUT:(DOWN) PRG DA CARICARE " :B:1-LEN(B):IFL=ORL>16ORL=OTHENRUN
1200 OPEN:1,8,15,"*:OPEN8,8,8,"0":+AS":.P.W":INPUT#1,A:IFAK>OTHENRUN:270
1209 :
1210 :
1211 :
1212 :REM*** SCRIVE LOW & HIGH BYTE
1214 :REM*** PER IL CARICAMENTO (.8,1)
1216 :REM*** E LA ROUTINE L.M.
1217 :REM*** DELL'AUTOSTART
1218 :
1219 :
1220 PRINT#8,CHR$(0);CHR$(1);:FORI=1TO259:PRINT#8,CHR$(2);:NEXT
1240 FORI=1TO72:READA:IFA=256THENA=L
1250 PRINT#8,CHR$(A);:NEXTI
1252 :
1256 :REM*** SCRIVE IL NOME DEL FILE
1257 :REM*** DA CARICARE
1258 :
1260 PRINT#8,B;CHR$(0);CHR$(0)
1270 INPUT#1,AS,BS,CS,DS:PRINT:(DOWN) (DOWN) " :AS,BS,CS,DS:" (DOWN) (DOWN) " :CLOSE:1,C
LOSEB
```

# MISCO

## 101 SOLUZIONI PER Con il catalogo Misco voi



**Tutti i nostri prodotti da voi...  
il giorno dopo!**

- Moduli continui • Floppy Disk
- Accessori terminali • Cavi
- Datacommunication • Nastri stampanti • Supporti magnetici
- Kit pulizia • Etichette • Mobili Workstation • Prevenzione antistatica



**A corto di spazio?**

Con l'ingegnoso Micro Fold Printerstand potete tenere la vostra stampante, i moduli continui e i tabulati uno sull'altro, in un unico compatto supporto.



**Urgenza di nastri e margherite?**

Abbiamo nastri per oltre 700 tipi di stampanti e un'ampia scelta di margherite. Sempre disponibili, tutti con consegna il giorno dopo.

**Floppy Disk troppo cari?**

Noi vi offriamo le migliori marche (3M, Olivetti, Verbatim, Memorex e molte altre) con garanzia completa e consegna immediata. A partire da sole 3.300 lire.



**Posizioni di lavoro scomode?**

Il nostro Monitor Stand ha due piani separati, entrambi regolabili per il miglior posizionamento della tastiera e del monitor. Se vi è più comodo, potete lavorare anche in piedi. Il tutto a 182.000 lire.



**Problemi di controllo sul computer?**

Il nostro esclusivo Tester kit (con 11 diversi tipi di tester), ha tutto quello che vi può servire per effettuare i vostri test sui collegamenti in pochi secondi, senza chiamare un tecnico.

# IL VOSTRO COMPUTER. SUBITO. ordinate oggi, noi consegnamo domani.



## Occhi arrossati, mal di testa, affaticamento?

Ecco i disagi più frequenti di chi lavora al video. Uno schermo antiriflesso e antistatico è la risposta a tutti questi problemi. A partire da 103.000 lire.



## Monitor ingombranti sulla scrivania?

Risolvete il problema con il braccio orientabile Misco Glideaway. Sostiene anche grandi monitor e, volendo, la stessa tastiera.

# NUOVO!

## Per chi lavora col computer

Finalmente anche in Italia c'è una grande organizzazione interamente al servizio di chi lavora col computer! A pochi mesi dalla pubblicazione del primo numero, si contano ormai a migliaia le Aziende che richiedono il catalogo Misco per avere in **24 ore** (massimo 48) gli articoli delle migliori marche e le novità più esclusive ai prezzi più vantaggiosi. E voi, vi siete già assicurati la vostra copia **gratuita** del catalogo Misco?

**MISCO**  
An affiliate of  
The Gillette Company



Richiedete il catalogo Misco telefonando oggi stesso

**02/900151**

oppure inviando questo coupon a: Misco Italy Computer Supplies S.p.A. - Il Girasole U.d.V. 2-01 - 20084 Lacchiarella Milano

H 4 1 4

**Sì**, inviate GRATIS il catalogo Misco a:

Azienda \_\_\_\_\_

Settore \_\_\_\_\_

Via \_\_\_\_\_ N. \_\_\_\_\_

CAP \_\_\_\_\_ Località \_\_\_\_\_

Prov. \_\_\_\_\_ Telefono \_\_\_\_\_

Marca e modello del computer \_\_\_\_\_

Inviare all'attenzione di:

Nome \_\_\_\_\_

Cognome \_\_\_\_\_

Posizione all'interno dell'Azienda \_\_\_\_\_

**MISCO**  
CATALOGO N. 1

**NOVITA'!**  
Accessori per il computer... con consegna immediata!

**PRODOTTI ESCLUSIVI!**  
EIP monitor, nuovi nostri scanner, Faster II per elaborare ogni tipo di controllo sui conteggi del vostro computer. Guardate a pag. 46

**MARCHE LEADER!**  
Se vi interessa la consegna in 24 ore di fiducia delle migliori marche: IBM, Olivetti, Maschun, Oxycom, Verbatim, Memorex leggete il catalogo a pag. 28

**LINEE ESCLUSIVE**

**GRATIS**

**VOI CHE VOLETE... NOI CONSEGNAMO DOMANI!**  
Controllate le nostre condizioni a pag. 55

Tutti gli accessori per il computer... con consegna immediata!

