

La programmazione del Mac La tecnica di stampa (2)

di Raffaello De Masi

La volta scorsa abbiamo visto che cosa era e come funzionava un driver di stampa. Ma, per dirla alla Lubrano, «la domanda nasce spontanea: ma è proprio necessario passare attraverso il driver di stampa, Apple o non?». La risposta è «certamente no!». Anzi probabilmente è più illogico dal punto di vista di un programmatore, passare attraverso un driver che scaricare tutto sulla porta di connessione.

La domanda è ancora più comprensibile se si tiene conto che, utilizzando il driver Apple, non si può disporre della stampa a media e alta qualità della pur eccellente Imagewriter II. Niente paura; è sufficiente bypassare il driver stesso per ritornare ai buoni vecchi codici di controllo, descritti anche nella più scalagnata e modesta stampante da un paio di biglietti da centomila. Eccoci quindi in arena aperta, senza più scudi e servitori che ci separano dalla parte più brutta della macchina

Il dumping di caratteri ASCII

A cosa può servire bypassare il driver? Ad una serie di cose, prima tra tutte l'invio diretto di file ASCII alla stampante. È ovvio che, con questa tecnica, il listato di un programma «fila» a velocità almeno doppia. Questa tecnica è, in ogni caso, non priva di sue caratteristiche personali, riassumibili come segue.

- Il dumping di testo scarica solo testo. Questo assioma tautologico non fa altro che affermare, se ce ne fosse bisogno, che la stampante associerà a ogni codice ASCII il carattere suo proprio. Non ci sono quindi possibilità di selezionare stili, grandezze o tipi particolari di carat-

- Questa tecnica è difficilmente adottabile su una stampante PostScript; se si esegue un output di quelli che vedremo appena appresso verso una stampante che interpreta tale linguaggio non accade assolutamente nulla. Se proprio si desidera abbracciare tale croce occorre disporre di AppleTalk e il testo deve essere interpretabile come un codice PostScript. Si tratta di una tecnica non facile da imparare, anche se ha dalla sua la estrema velocità di trasferimento dei dati.

Ma come si fa a inviare effettivamente un file testo alla stampante nel modo appena detto? Proviamo con QuickBasic; il codice minimo in tale ottica è:

```
OPEN "LPT1:DIRECT" FOR OUTPUT AS #7
PRINT#7, "MCmicrocomputer, la scelta professionale!"
CLOSE#7
```

tere (ameno di quelli presenti sulle ROM della stampante stessa).

- Poiché si stanno riversando codici direttamente su una delle porte seriali, occorre conoscere fisicamente dove è la stampante e dirigere, mediante opportuni codici, la scrittura verso quella porta stessa.

- Poiché non interviene in alcun modo il driver della stampante, è possibile, co-

dove #7 indica un buffer-canale di stampa. Ovviamente, proprio perché non si attinge a un driver di stampa, il comando [WINDOW OUTPUT] è inutile, anzi dannoso.

In ZBasic il comando è abbastanza simile, anche se il manuale presenta un errore di stampa che pregiudica il risultato. Il sorgente analogo a quello precedente è:

```
rem -2 per la porta stampante, -1 per la porta modem
Messaggio$ = "MCmicrocomputer, la scelta professionale!"
OPEN"C", -1, 9600 ' -2 se si usa la porta stampante
HANDSHAKE -1,-1
PRINT #-1, Messaggio$
CLOSE#1
```

me dicevamo prima, trasferire codici di controllo alla stampante ricevente.

Ovviamente questa tecnica non è priva di pericoli, se si considera che essa è strettamente dipendente dalla stampante adottata; come tale non è certo adottabile senza preoccupazioni se si desidera distribuire la nostra applicazione a una vasta fascia di utenza.

Per concludere questo argomento, possiamo senz'altro dire che il dumping del testo è effettivamente semplice, ma non ha mai avuto grande fortuna e rimane a livello di curiosità. Da una parte non è possibile configurare la nostra applicazione, specie se sarà ampiamente distribuita, per tutte le stampanti in circolazione, inoltre nessun possessore

Mac è tanto folle da comprare una stampante che non disponga di un driver dedicato. Non a caso Mac è un mondo a parte, e, almeno dal mio punto di vista, un mondo migliore.

Stampiamo da Quick Basic

Passata la curiosità del dumping di testo, ritorniamo all'ambiente Mac per studiare la migliore tecnica di trasferimento del testo dalla nostra applicazione alla carta.

Sebbene Mac sia una macchina a sé stante e per molti versi migliore di tutta la concorrenza, ha un patrimonio da rispettare e da trascinare, se vogliamo. Poiché i primi interpreti e compilatori dei diversi Basic Microsoft erano costruiti per macchine IBM o comunque MS-DOS, a causa di ciò le versioni Mac del Microsoft Basic, sia nella versione tout court, sia in quella Quick Basic, hanno una serie di caratteristiche, ereditate dal mondo MS-DOS, che poco si adattano alle caratteristiche proprie di questa macchina. Un esempio è quello, appunto della stampa, con le sue routine di indirizzamento.

Sotto MS-DOS le differenti porte di comunicazione o di stampa possono essere raggiunte aprendo (OPEN) la periferica come se fosse un file dati. Le porte seriali sul PC sono etichettate come COM1: e COM2:, le porte parallele, cui generalmente si connette una stampante sono etichettate LPT1: e LPT2: ; in tutti i casi i due punti fanno parte dell'etichetta.

In ambiente Mac l'accesso alla stampante avviene in modo del tutto simile a quello PC; è sufficiente pensare per un momento di lavorare su un IBM e aprire la periferica di stampa con il classico [OPEN LPT1:]. Il fatto strano è che su Macintosh non esistono porte parallele; sia la Imagewriter che la Laserwriter sono periferiche seriali. E allora? Ecco che si ritorna al significato originario della sigla, Line Printer 1, e tutto ritorna chiaro.

Con Mac occorre infine passare attraverso due finestre di dialogo, il PageSe-

tup e il PrintJob, ben note a tutti gli utenti. Sfortunatamente, in QuickBasic (ma non in ZBasic) non c'è la possibilità di separare i due tipi di dialogo, magari per chiamarli in momenti differenti. Inoltre (speriamo che questi difetti saranno sanati nella prossima versione) non esiste possibilità di verificare se è stato premuto il pulsante CANCEL. La sintassi per chiamare i driver di stampa è la seguente:

```
OPEN "LPT1:PROMPT" FOR OUTPUT AS #7
```

Si raccomanda di non lasciare spazi tra LPT1[:] e PROMPT, in quanto non si tratta di comandi separati ma di un unico comando. Piccola considerazione è che questo comando chiama prima il driver di stampa e successivamente apre l'indirizzo di stampa, assegnandogli il n° 7 (o, ovviamente, qualunque altro).

Una volta «aperta la strada» della stampante, è necessario indirizzare l'output in questa direzione. Questo avviene col comando:

```
WINDOW OUTPUT #7
```

dove ovviamente, il numero 7 deve coincidere con quello presente nel comando OPEN.

Dopo di ciò non è più compito del programmatore preoccuparsi di altre problematiche relative alla stampa; ogni scritto, disegno, o comunque comando grafico è inviato alla stampante; e, in ossequio alla più stretta sintassi dei file, il flusso può essere interrotto con il comando CLOSE.

Ciononostante non è tutto così facile. Occorre invece tenere a mente certe considerazioni, peraltro intuitive. Ad esempio, sebbene la stampante si comporti, per quanto riguarda la grafica, né più né meno che come uno schermo, certi comandi possono non funzionare, o funzionare in maniera imprevedibile. Un esempio sono quelli di refreshing dello schermo, o addirittura quello di [CLS], che può dare risultati inattesi se

inserito nel corso di un output. Ancora il comando LPRINT, utilizzabile con soddisfazione su Imagewriter, non funziona con la laser. Infine, se si schiaccia il comando CANCEL nel PrintJob, il nostro programma, all'oscuro di quanto sta succedendo, tenterà di continuare con il comando successivo a quello che comprende il WINDOW OUTPUT #7. Poiché questo «file» non è stato mai aperto, si genera una condizione d'errore (error 52 — Bad File Number), lo stesso errore in pratica (se ci si pensa un momento ci si rende conto che si tratta di qualcosa di completamente analogo) di quello che avviene quando non si è eseguito l'indirizzamento di porta con l'opzione «Scelta Risorse» di DA. È questo il motivo per cui occorre creare una routine di trap d'errore durante la stampa.

Come costruire questa routine in modo che funzioni in maniera efficiente e sicura è cosa non difficile. La prima cosa da fare, ovviamente è quella di creare un menu specifico di stampa.

Le prime righe del nostro programma saranno, quindi:

```
MENU 1,0,1, "Archivio"
MENU 1,1,1 "Stampa" : CmdKey 1,1,"P"
MENU 1,2,0, "-----"
MENU 1,3,1, "Esci" : CmdKey 1,3,"Q"
```

Occorre poi modificare il MenuHandler aggiungendo ad esso un comando CASE (o simile, come un LONG IF) del tipo:

```
ControlloMenu:
NMenu = MENU(1) 'trap del tipo di menu
SELECT CASE NMenu
CASE 1 : GOSUB Stampa
'CASE 2 : non esiste
CASE 3 : END
CASE ELSE
BEEP
END SELECT
MENU 'ripristina il menu vuoto
RETURN
```

È il caso adesso di costruire le routine

cui si è appellato il menu stesso. Vediamo quella di stampa che abbiamo etichettata come appunto [Stampa]

```
'XXXXXXXXXXXXXXXXX
'Routine di stampa del contenuto della finestra
'XXXXXXXXXXXXXXXXX
Stampa:
var1 = WINDOW(1)
ON BREAK GOSUB maneggio_break
ON ERROR GOTO maneggio_errore
```

```
' routine di break di stampa
'reagisce alla combinazione COMMAND-. interrompendo la stampa
break_di_stampa:
RETURN Uscita_Stampa
```

va routine di indirizzamento, già vista, modificandola come segue:

```
n# = FRE(-1) 'cancella la memoria inutilmente occupata
WINDOW var1
OPEN "LPT1:PROMPT" FOR OUTPUT AS #7
CURSOR = 4 'trasforma il cursore in orologio, durante la stampa
WINDOW OUTPUT #7
GOSUB Disegno 'ovviamente questa chiamata puo' essere sostituita
'dai diretti comandi di output
CLOSE
```

La prima cosa da fare è settare la variabile var1 come numero dell'attuale finestra attiva in modo da essere utilizzata, in caso di chiamata, per disegnare il contenuto della finestra stessa. Gli altri due indirizzamenti del programma sono abbastanza ovvi in quanto con essi invieremo immediatamente a routine di maneggio di break e di errore. Al trap d'errore si accede quando si preme il pulsante di CANCEL (abbiamo appena fatto notare come l'uso di questo generi immediatamente un errore 52). Inoltre c'è da tener conto di un fatto, usando una laser; poiché il driver di

Occorre ricordare che tra il comando [WINDOW OUTPUT #7] e CLOSE non ci possono essere altri comandi di output di WINDOW, a scanso di errori imprevedibili. Generalmente la routine di break e di errore è specifica per la stampa, ed è quindi diversa da quella/e pur necessaria del programma principale. Dopo l'esecuzione della routine di stampa occorrerà rimettere tutto in ordine, restituendo il controllo alle routine principali, con una pièce di programma del tipo:

```
Fine_stampa:
' ripristino condizioni principali
```

```
ON BREAK GOSUB Maneggio_break_principale
ON ERROR GOTO 0 'o GOTO in altro posto, se esiste una routine apposita
INITCURSOR " ritrasforma il cursore se si e' utilizzato CURSOR 4
CLOSE
RETURN
```

questa richiede un certo quantitativo di memoria, QBasic ha bisogno di fare spazio e di eseguire un poco di garbage collection. Questo avviene attraverso il comando FRE(-1) che costringe l'applicazione a calcolare la quantità di memoria disponibile, compattando la stessa e rendendo disponibili le aree cancellabili.

Bene, adesso occorre assicurarsi che la nostra finestra di output sia anche quella attiva, chiamando la routine di disegno della pagina immediatamente dopo. Utilizzeremo, quindi la nostra bra-

E, tanto per completare, codifichiamo anche il nostro trap d'errore:

```
Trap_errore:
errore1 = ERR
CLOSE
IF errore = 52 THEN RESUME Uscita_stampa
'si possono usare altri IF errore = ...
BEEP
RESUME Uscita_stampa
```

E proprio, come raffinatezza = ciliegia sulla torta, possiamo addirittura maneggiare il trap d'errore durante la stampa. Battiamo:

È davvero tutto! Ovviamente la chiamata [Disegno] conterrà tutto l'output grafico relativo.

Riassumendo

Riassumendo, una serie di cose, da fare e da non fare, nell'output di stampa.

Da fare:

- Usare il comando OPEN "LPT1: PROMPT" FOR OUTPUT AS #7, per dirigere la stampa, avendo cura di usare un numero alto di riferimento, in modo da non interferire con il maneggio dei file dati. Usate strettamente la sintassi proposta, ricordando tra l'altro di non lasciare spazi non richiesti o non specificati.

- Usare il comando WINDOW OUTPUT # per reindirizzare l'uscita di stampa.
- Disinserire tutti gli «event trapping» e le funzioni di «timer trapping» prima della stampa; specie nel secondo caso la lentezza della stampa può interferire pesantemente con l'esecuzione del programma.

Da non fare:

- Utilizzare un nuovo WINDOW OUTPUT prima del termine della stampa; può essere utile, talvolta, inserire un ritardo nella prosecuzione del programma fino a quando non si è scaricato completamente il buffer.

- Utilizzare TAB per la formattazione della stampa. TAB non è un comando ben costruito e può portare a errori di disallineamento in caso di scrittura proporzionale. È preferibile, e talvolta necessario, utilizzare il comando MOVE-TO.

- Lasciare un trap d'errore senza usare il comando RESUME. Fino a quando un programma scorre senza trovare il RESUME, si mantiene sempre in trap d'errore; ora se si lascia la stampa, senza aver «riassunto», un nuovo errore tenterà di eseguire l'escape d'errore previsto per la stampa, con le conseguenze immaginabili.

Ma, per quanto attiene alla stampa, non abbiamo certo finito; ci risentiremo la prossima volta.

CONFIGURA IL TUO PC!

CABINET

| | |
|---|-----------|
| <input type="checkbox"/> Desktop + alimentatore 200Watt | £ 140.000 |
| <input type="checkbox"/> Minitower LCD + alimentatore 200Watt | £ 250.000 |
| <input type="checkbox"/> Tower LCD + alimentatore 230Watt | £ 350.000 |

PIASTRE MADRI

| | |
|---|-------------|
| <input type="checkbox"/> 286 16-21Mhz | £ 190.000 |
| <input type="checkbox"/> 286 20-25Mhz | £ 290.000 |
| <input type="checkbox"/> 386 sx 20-25Mhz | £ 490.000 |
| <input type="checkbox"/> 386 dx 33-50Mhz | £ 990.000 |
| <input type="checkbox"/> 386 dx 40-70Mhz | £ 1.290.000 |
| <input type="checkbox"/> 486 sx 25-100Mhz | £ 1.590.000 |
| <input type="checkbox"/> 486 dx 33-160Mhz | £ 2.490.000 |

MEMORIA RAM

| | |
|---|-----------|
| <input type="checkbox"/> RAM 1 Megabyte | £ 100.000 |
| <input type="checkbox"/> RAM 2 Megabyte | £ 190.000 |
| <input type="checkbox"/> RAM 4 Megabyte | £ 350.000 |
| <input type="checkbox"/> RAM 8 Megabyte | £ 690.000 |

DISCHI FISSI

| | |
|--|-------------|
| <input type="checkbox"/> 40 Mb AT-BUS cache 19 ms | £ 390.000 |
| <input type="checkbox"/> 80 Mb AT-BUS cache 17 ms | £ 590.000 |
| <input type="checkbox"/> 130 Mb AT-BUS cache 17 ms | £ 790.000 |
| <input type="checkbox"/> 210 Mb AT-BUS cache 13 ms | £ 1.490.000 |

SCHEDE GRAFICHE

| | |
|---|-----------|
| <input type="checkbox"/> VGA 16 bit 256 kb (800 x 600 x 16 colori contemporanei) | £ 100.000 |
| <input type="checkbox"/> VGA 16 bit 512 kb Tseng Labs ET 3000 (1024 x 768 x 16 colori contemporanei) | £ 150.000 |
| <input type="checkbox"/> VGA 16 bit 1 Mb Tseng Labs ET 4000 (1024 x 768 x 256 colori contemporanei) | £ 250.000 |

MONITOR

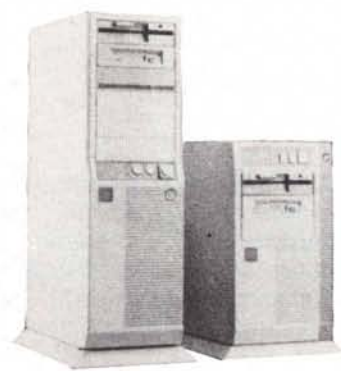
| | |
|---|-------------|
| <input type="checkbox"/> Monocromatico VGA 14" fosfori bianchi | £ 195.000 |
| <input type="checkbox"/> Colori super VGA 1024 x 768 14" 0.28 dot pitch | £ 595.000 |
| <input type="checkbox"/> Colori super VGA 1024 x 768 19" | £ 1.950.000 |

KIT OBBLIGATORIO

| | |
|---|-----------|
| <input type="checkbox"/> Comprendente 1 disk drive 3.5" 1.44 Mb formattati, controller hard disk e floppy disk AT-BUS interleave 1:1, 2 porte seriali, 1 porta parallela, 1 porta joystick, mouse Microsoft compatibile, tastiera italiana estesa, software di emulazione coprocessore matematico, contributo spese di assemblaggio e collaudo della configurazione scelta, garanzia 12 mesi su tutti i componenti. | £ 290.000 |
|---|-----------|

ISTRUZIONI PER L'USO

Grazie a questo semplice schema, si ha la possibilità di configurare un personal computer in base alle proprie esigenze, senza essere vincolati a modelli prefabbricati; si possono ottenere migliaia di combinazioni, semplicemente scegliendo: 1 Cabinet, 1 Piastra Madre, un tot di Memoria RAM, 1 Disco Fisso, 1 Monitor e l'indispensabile Kit obbligatorio. Sommando gli importi dei singoli componenti, si ottiene il prezzo totale che sarà una piacevole sorpresa.



OPZIONI E ACCESSORI

(la cui assenza non compromette il corretto funzionamento del Personal Computer)

| | |
|---|-----------|
| <input type="checkbox"/> Disk drive aggiuntivo (5.1/4" 1.2Mb) | £ 120.000 |
| <input type="checkbox"/> Coprocessore matematico 80287 | £ 190.000 |
| <input type="checkbox"/> Coprocessore matematico 80387 sx | £ 290.000 |
| <input type="checkbox"/> Coprocessore matematico 80387 dx | £ 490.000 |
| <input type="checkbox"/> Streaming Tape 250 Mb | £ 795.000 |
| <input type="checkbox"/> Tavoleta grafica A4 (12"x12") | £ 490.000 |
| <input type="checkbox"/> Tavoleta grafica A3 (18" x 12") | £ 790.000 |
| <input type="checkbox"/> Scanner manuale 400 Dpi B/N | £ 290.000 |
| <input type="checkbox"/> Scanner manuale 400 Dpi a colori | £ 750.000 |
| <input type="checkbox"/> Modem interno 2400 Baud V21/V22 | £ 190.000 |
| <input type="checkbox"/> Scheda Fax 9600 Baud | £ 495.000 |
| <input type="checkbox"/> Scheda audio Sound Blaster | £ 295.000 |
| <input type="checkbox"/> Digital Research DR DOS 5.0 | £ 100.000 |

prezzi IVA esclusa Le caratteristiche tecniche possono variare senza preavviso.



DESENZANO (BS) PIAZZA MALVEZZI,14 TEL.030/9911767
 BRESCIA CORSO MAGENTA,32/B TEL.030/3770200
 VERONA PIAZZA S.TOMMASO,10/11 TEL.045/8010782
 GRUMELLO (BG) VIA ROMA,61 TEL.035/833097