

# Assembler 68000

di Marco Pesce

*Terminato il discorso sul 68000 e le sue istruzioni cominciamo a rivolgere la nostra attenzione al sistema operativo di Amiga, al fine di mettere realmente in pratica le conoscenze acquisite e realizzare qualcosa di concreto e magari anche utile*

Diamo uno sguardo alle risorse messe a nostra disposizione. Il sistema operativo del nostro computer è suddiviso in «librerie» ovvero in gruppi di subroutine specializzati in determinati settori, così abbiamo una libreria per la grafica, una per i suoni e così via. Per usufruire delle funzioni di una libreria occorre «aprirla» ovvero comunicare al sistema operativo le nostre intenzioni, che prontamente ci fornirà un apposito puntatore da utilizzare unitamente a semplici «scostamenti» noti, per effettuare le nostre chiamate, il tutto nell'intenzione di evitare il gravoso compito di conoscere la effettiva allocazione nelle ROM delle varie routine. In pratica il puntatore che ci viene fornito si riferisce a un sistema di istruzioni JMP che contengono i salti assoluti alle routine; ogni istruzione JMP occupa 6 byte; quando si effettua una chiamata con un'istruzione del tipo "JSR -30(A6)" viene eseguita una di queste JMP, nella fattispecie, la prima; lo scostamento indica ovviamente qual è la JMP da eseguire e infatti va di 6 in 6 byte, quindi con JSR -36(A6) verrà eseguita la seconda routine e così via. Nel registro indirizzi A6 va ovviamente memorizzato il puntatore fornito all'apertura della libreria. Solo una delle diverse librerie è immediatamente disponibile; EXEC. Il suo puntatore è memorizzato nella locazione Ram numero 4 (è una longword) ed è quindi questo il valore da depositare nel registro A6 per usufruire delle sue routine. I vari scostamenti sono tabulati e in teoria vengono sostituiti automaticamente al nome della routine, ma io preferisco usarli direttamente. Le routine di Exec permettono la gestione delle librerie e del multitasking di Amiga, con relativi

elementi di contorno. Tali routine sono di concezione molto avanzata e sarà bene affrontarle in un secondo momento. Per ora impariamo ad aprire una libreria.

Per aprire una qualunque libreria occorre utilizzare la routine di Exec "openlibrary". Nel registro A1 va depositato il puntatore al nome della libreria, ovvero a una struttura dati che contiene i caratteri in codice ASCII appunto del nome, realizzabile con la seguente riga:

```
NOME DC.B 'graphics.library',0
```

lo zero indica la fine del nome. Nel registro D0 va depositato il valore che indica la versione della routine da utilizzare. Preparati questi due registri possiamo effettuare la chiamata. Il risultato dell'operazione è il puntatore alla libreria che noi trasferiamo dal registro D0 in un'opportuna locazione tampone. Se disgraziatamente in tale registro troviamo il valore 0 vuol dire che non possiamo aprire questa libreria (non dovrebbe succedere mai di norma). Il listato di figura 1 esemplifica quanto detto.

Dal momento che la libreria aperta è quella grafica vediamo alcune sue routine. Un primo raggruppamento è quello che permette la definizione della struttura di uno schermo grafico, ovvero le sue dimensioni, il suo posizionamento, i suoi colori, il suo modo grafico, ecc. Aprire uno schermo con tali routine non è un'impresa semplicissima e nella maggior parte dei casi conviene usufruire di Intuition, ovvero di un'altra libreria, quella che gestisce le varie finestre grafiche (e relativi gadget) presenti anche nel workbench. Invece di perdere tempo ad aprire lo schermo con le routine della graphics library utilizziamo Intuition per poi dedicarci in tutta tranquillità alle routine grafiche; il listato di figura 2 effettua l'apertura della libreria e di uno schermo in 320x200 32 colori, inoltre vengono memorizzati gli indirizzi di alcune importanti strutture dati che verranno utilizzate in seguito.

Aperto il nostro schermo vediamo di disegnarci qualcosa. Ci occuperemo di: come disegnare linee e rettangoli pieni, come cambiare colori nella palette, come colorare e come leggere un pixel.

La routine «Draw» permette di dise-

## Apertura di una libreria

```
MOVE.L 4,A6
MOVEQ #0,D0
LEA NOME(PC),A1
JSR -552(A6); -552 e' lo scostamento per la OPENLIBRARY
MOVE.L D0,GLIB
```

```
NOME DC.B 'graphics.library',0
GLIB DC.L 0
```

Figura 1

gnare linee sullo schermo; le coordinate di partenza sono quelle dell'ultimo pixel disegnato, mentre le coordinate di arrivo vengono stabilite al momento della chiamata. Altra cosa da stabilire è la rastport nella quale bisogna eseguire l'operazione. Se vogliamo stabilire anche le coordinate di partenza della linea, quindi non ci accontentiamo di continuare il disegno dalla vecchia posizione del cursore basta usufruire della routine «Move» che permette di riposizionarlo a piacimento. Anche in questo caso occorre specificare la rastport oltre, ovviamente, alle coordinate della nuova posizione del cursore.

Ecco un esempio di utilizzo delle due routine: ▼

```
MOVE.L GLIB,A6
MOVE.L POSX,D0; coordinata X finale
MOVE.L POSY,D1; coordinata Y finale
MOVE.L RASTPORT,A1
JSR -246(A6); routine DRAW
```

```
MOVE.L GLIB,A6
MOVE.L NPOSX,D0; nuova coordinata X
MOVE.L NPOSY,D1; nuova coordinata Y
MOVE.L RASTPORT,A1
JSR -240(A6); routine MOVE
```

Spero che a questo punto il meccanismo cominci ad esservi chiaro. Per stampare un rettangolo ad ultravelocità basta servirsi della routine che sfrutta a pieno le capacità del blitter in questo campo (tra l'altro anche la Draw usufruisce del blitter) ovvero la Rectfill. Il «delta» è «-306»; nel registro A1 va depositato il puntatore alla rastport, in D0 la coordinata x minore, in D1 la y minore e in D2 e D3 rispettivamente la x e la y maggiore.

Nulla abbiamo detto nei confronti del colore da utilizzare per la stampa né di come si modificano quelli standard delle palette.

Per settare i colori dell'intera palette si utilizza la LoadRGB4, che necessita di una struttura contenente una tabella con i colori da modificare. La tabella deve essere una sequenza di valori per un totale di 64 byte nel caso si utilizzi l'intera palette, quindi 2 byte per colore. Un colore si definisce in base al dosaggio delle tre componenti fondamentali ovvero rosso, verde e blu. I primi 4 bit del byte meno significativo (il secondo) indicano la componente di blu, i secondi indicano quella di verde e i 4 bit meno significativi dell'altro byte indicano la componente di rosso (ogni componente può oscillare quindi tra 0 e 15). L'insieme di queste tre combinazioni ci permette di formare uno qualunque dei 4096 colori disponibili (\$000 nero \$FFF bianco). Per chiamare la LoadRGB4 occorre depositare in A0 la viewport dello screen in A1 il puntatore alla tabella

```
MOVE.Q #0,D0
MOVE.L 4,A6
LEA NOME2(PC),A1
MOVE.L D0,ILIB
MOVE.L ILIB,A6
LEA DATI(PC),A0
JSR -198(A6)
MOVE.L D0,PSCREEN
MOVE.L GLIB,A6; vedi listato fig.1
MOVE.L PSCREEN,A0
ADD.L #44,A0
MOVE.L A0,VIEWPORT
ADD.L #40,A0
MOVE.L A0,RASTPORT
MOVE.L VIEWPORT,A4
ADD.L #4,A4
MOVE.L (A4),COLORMAP
MOVE.L COLORMAP,A4
MOVE.L (A4),COLORTAB
MOVE.L RASTPORT,A4
ADD.L #4,A4
MOVE.L (A4),BITMAP
MOVE.L BITMAP,A4
ADD.L #8,A4
MOVE.L (A4),BITP1
MOVE.L BITMAP,A4
ADD.L #12,A4
MOVE.L (A4),BITP2
MOVE.L BITMAP,A4
ADD.L #16,A4
MOVE.L (A4),BITP3
MOVE.L BITMAP,A4
ADD.L #20,A4
MOVE.L (A4),BITP4
MOVE.L BITMAP,A4
ADD.L #24,A4
MOVE.L (A4),BITP5
RTS

NOME DC.B 'intuition.library',0
ILIB DC.L 0
DATI DC.B 0,0,320,200,5,1,64,$10F,0,0,0,0,0,0,0,0
PSCREEN DC.L 0
VIEWPORT DC.L 0
RASTPORT DC.L 0
COLORMAP DC.L 0
COLORTAB DC.L 0
BITMAP DC.L 0
BITP1 DC.L 0
BITP2 DC.L 0
BITP3 DC.L 0
BITP4 DC.L 0
BITP5 DC.L 0
```

Figura 2

(colormap) e in D0 un contatore che indichi il numero di colori da modificare (32 nel nostro caso); il delta è «-192». Per selezionare uno dei 32 colori per la nostra «penna» ovvero per il colore da utilizzare nel disegno si usa la routine SetApen, che come dice il nome seleziona il colore per la penna A (quella utilizzata dalla Draw e dalla Rectfill); il delta è «-342». In A1 va la rastport e in D0 il numero d'ordine corrispondente nella palette colore.

Passiamo infine alla routine per disegnare un singolo pixel e per leggere il colore di un particolare pixel sullo schermo. La prima routine usufruisce del registro A1 per specificare la rastport e dei registri D0 e D1 per, rispettivamente, la coordinata x e la coordinata y (il colore è quello della penna A); il delta è «-324». Per leggere si usa la stessa

combinazione di registri ma il delta è di «-318» e in D0 viene restituito il colore utilizzato in quel pixel (che può ovviamente anche essere lo sfondo, ovvero il colore 0).

Ultima routine per questo mese, quella che cancella lo schermo partendo dall'ultima posizione occupata dal cursore fino alla fine della pagina grafica; il delta è «-48» mentre il registro da settare è soltanto l'A1 con la rastport. Con questo insieme di routine potete esercitarvi e «giocare» quanto basta con la grafica, magari tentando di ottenere paesaggi artistici col l'ausilio di funzioni matematiche o cose del genere, il tutto nell'intenzione di familiarizzare con le routine e di mantenersi in allenamento con le istruzioni in LM. Arrivederci.

MC