

PROVA



Add-on per Clipper 5 Blinker, Funcky e DGE

di Gabriele Romanzi

Il successo di un pacchetto software viene spesso decretato dalla quantità di prodotti accessori per esso disponibili; seguendo questo metro dovremmo collocare il Clipper ai primi posti di una ipotetica classifica in quanto, sin dalle sue prime versioni, è stato sempre supportato da terzi produttori di librerie e tool di sviluppo.

Il Clipper, uno dei più famosi prodotti per la gestione di basi di dati (e non solo) in ambiente MS-DOS, ha ormai raggiunto con la nuova release 5 un ruolo di primissimo piano tra i sistemi di sviluppo per PC; da semplice compilatore di procedure realizzate in DBIII si è via via trasformato in un potentissimo linguaggio di programmazione struttura-

to, che poco o niente ha da invidiare ad altri linguaggi come il C, con il quale è peraltro interfacciabile, permettendo così un'ampia flessibilità di utilizzo.

L'uscita sul mercato dell'attesissima nuova versione di questo compilatore (alla cui prova apparsa sul numero 103 vi rimando) ha avuto come conseguenza la necessità da parte dei produttori di pacchetti ausiliari di aggiornare i loro cataloghi con versioni specifiche per la versione 5, che differisce dalle precedenti non tanto per le funzionalità specifiche del linguaggio quanto per una sostanziosa ristrutturazione della gestione della memoria, delle variabili e degli operatori oltre che per l'introduzione di alcune classi di oggetti, preludio di una

versione object-oriented che voci di corridoio danno in fase di progettazione alla Nantucket.

Dalla società Algol di Milano, che cura la distribuzione del Clipper e dei suoi più famosi tool in Italia, abbiamo ricevuto le prime novità disponibili per la nuova versione, che coprono un ampio ventaglio di esigenze del programmatore, sia esso alle prime armi che un professionista: analizzeremo infatti, nel corso di questa prova, il Blinker (linker ad overlay dinamici), la Funcky Lib (libreria «general-purpose» tra le più note), la DGE (libreria grafica) e la NetLib (libreria per la realizzazione di procedure in rete locale).

Come vedete ce n'è per tutti i gusti e

penso che ognuno possa trovare il prodotto adatto alle sue specifiche esigenze, da integrare nelle applicazioni che sta sviluppando lì dove il Clipper possa dimostrarsi più o meno carente.

La prova, da quanto detto finora, non sarà assolutamente comparativa data la diversità di impiego dei vari prodotti, ma vorrà essere soprattutto una panoramica (certamente non esaustiva) di alcune tipologie di prodotti disponibili per il programmatore Clipper, sempre più alla ricerca di strumenti che gli permettano di utilizzare questo linguaggio di programmazione al massimo delle sue possibilità.

Add-on Clipper

Produttori:
DGE 4.0: Bits Per Second - (Inghilterra)
Funcy: Dlesko (USA)
Blinker: Blink Incorporated (Inghilterra)

Prezzi (IVA esclusa):
DGE 4.0 Libreria grafica L. 740.000
Funcy Libreria multifunzione L. 650.000
Blinker Dinamic Overlay Linker L. 760.000

Distributore:
Algol
Via Feltre, 28/6 - 20132 Milano
Tel. 02/26411411

Ma come, potrete chiedervi, abbiamo letto nella prova del Clipper 5 che la Nantucket ha adottato un nuovo linker, veloce e dalle molteplici possibilità, ed ora leggiamo la prova di un altro linker da acquistare a parte che lo dovrebbe rimpiazzare?

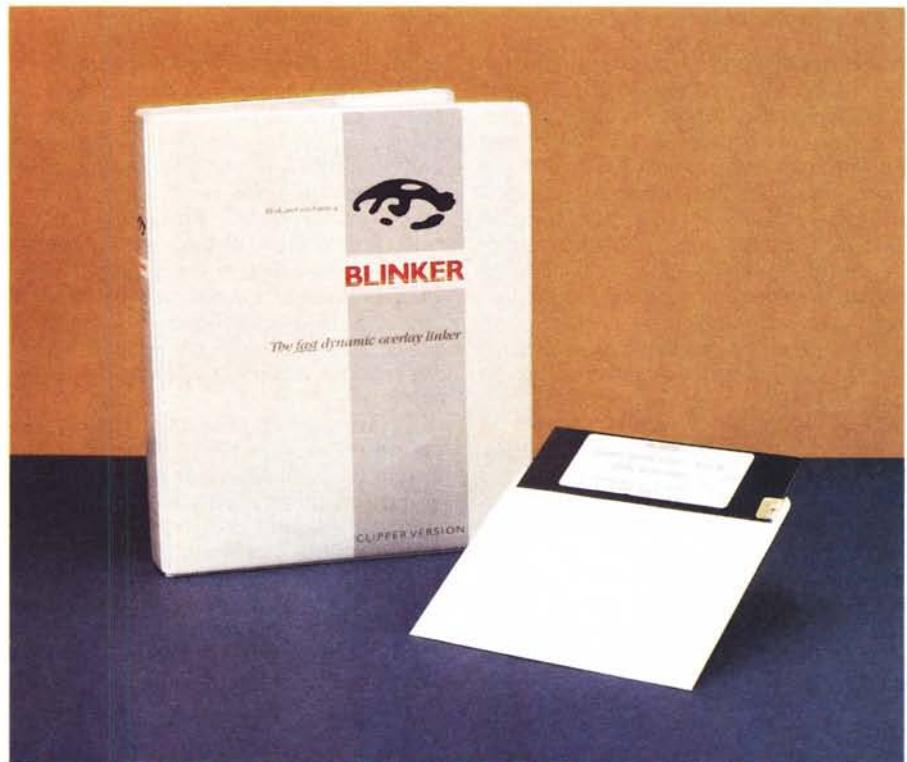
Si sa, nel mondo dell'informatica è sempre difficile detenere la palma del migliore in un certo campo per lungo tempo, ma nel caso del Blinker ci troviamo di fronte ad un prodotto che ormai da molti anni è diventato uno strumento inseparabile per molti programmatori Clipper.

Il Blinker è un linker ad overlay dinamici che ha conquistato negli ultimi anni un ruolo di primo piano tra gli add-on per Clipper; con la precedente versione di questo compilatore (Summer 87) veniva fornito un linker, il Plink 86, che, a parte la sua cronica lentezza, non permetteva di realizzare programmi che andassero oltre il famigerato limite dei 640 Kbyte imposto dal MS-DOS.

Occorreva quindi suddividere il programma in tanti moduli distinti (overlay) da caricare ciascuno in memoria soltanto quando necessario ed in maniera mutuamente esclusiva tra di loro, richiedendo un grande sforzo in fase progettuale dell'applicazione per evitare che più moduli si «pestassero i piedi» tra di loro.

Con il Blinker, invece, questi problemi vengono risolti facilmente in quanto è sufficiente comunicare al linker quali sono le procedure che devono essere gestite ad overlay ed il gioco è fatto: è il Blinker stesso, in maniera del tutto trasparente per l'utente, ad ottimizzare le allocazioni delle funzioni e procedure nei vari moduli di overlay ed a gestire il loro swapping dinamico.

Con l'overlay dinamico permesso dal Blinker il programmatore può quindi



Blinker

concentrarsi più sullo sviluppo dell'applicativo che sulla strutturazione dei suoi moduli componenti; inoltre, al momento dell'esecuzione del programma, l'«overlay manager» tiene traccia di quali e quanti moduli vengono utilizzati in ogni istante dal programma ed in caso ci sia bisogno di uno spazio di memoria mag-

giore di quello disponibile si occupa dello scarico temporaneo su memoria di massa degli overlay secondo uno schema di priorità che tiene conto della frequenza di utilizzo e della loro «storia passata».

Un altro punto a favore del Blinker rispetto ad altri linker per Clipper è la

```
B:\BLINKMAI.N\C50>blinker
(=) (=)      BLINKER Dynamic Overlay Linker 1.40.
  ^
  _      Blink and you'll miss it !!

Copyright (C) Blink, Inc., 1990. All Rights Reserved.
Serial # EB-006459. (Clipper 5.0 version).

BLINKER : 1102 : no input files were specified

B:\BLINKMAI.N\C50>_
```

Schermata del Blinker. Una simpatica faccina ci strizza l'occhio durante il link del programma.

possibilità di gestire con overlay dinamici sia proprie librerie di funzioni scritte in C ed Assembler, sia le librerie del Clipper quale la Extend.lib; è chiaro che funzioni particolari (quali quelle che coinvolgono l'uso di interrupt) possono essere soggette a restrizioni perché non garantiscono una sufficiente integrità dell'ambiente operativo in fase di esecuzione del programma, ma sostanzialmente tutte le funzioni che «si comportano bene» non hanno problemi ad essere gestite con overlay dinamici da questo linker.

Oltre a questa capacità notevole in termini di gestione dell'occupazione della memoria da parte di un programma, il Blinker è in grado di produrre un modulo eseguibile in un tempo sensibilmente inferiore a quello richiesto dai linker tradizionali (quali RTLink o MSLink); effettuando dei test su una procedura da me realizzata in Clipper 5 (circa 300 Kbyte di sorgenti oltre ad una libreria di più di 250 Kbyte) i tempi necessari al link si sono ridotti drasticamente con l'utilizzo del Blinker, arrivando ad attestarsi su una media di circa 15 secondi (dagli oltre due minuti necessari agli altri linker).

Se inoltre sfruttiamo la capacità che ha il Blinker di effettuare un link incrementale dei soli «.OBJ» modificati dall'ultima volta raggiungiamo delle prestazioni incredibili, dell'ordine di qualche secondo: una vera e propria «Ferrari» nelle mani del programmatore che guadagna tempo prezioso nella fase di sviluppo.

Già queste caratteristiche da sole permettono di capire come il successo di questo prodotto continui anche dopo l'uscita del Clipper 5 dotato del RTLink, che dalla sua mantiene comunque un

vantaggio: la possibilità di creare delle PLL (Pre-Link Library) in cui memorizzare parti di codice comuni a più moduli in modo da rendere minore la dimensione del file «.EXE» finale. L'eseguibile della procedura utilizzata per il test è salito infatti da circa 180 Kbyte (RTLink) a circa 500 Kbyte (Blinker) con l'inconveniente che mentre in precedenza riusciva a partire su un server di rete non dedicato (aveva bisogno di circa 300 Kbyte di RAM libera) nel secondo caso necessitava di uno spazio di gran lunga maggiore.

C'è comunque da spezzare una lancia (anzi due) a favore del Blinker per quanto riguarda la dimensione dell'eseguibile prodotto. Innanzitutto per effettuare una prova significativa sui tempi di link non ho creato uno script ottimizzato per il Blinker, ma ho utilizzato lo stesso del RTLink; in secondo luogo la copia ricevuta per il test comprendeva due versioni differenti a seconda che si utilizzasse la versione «Summer 87» o la più recente «5»: in quest'ultimo caso il livello di revisione del Blinker è 1.4, ma sul dischetto è già presente (sotto forma di file di testo) il manuale della versione 1.5 che sarà rilasciata subito dopo l'uscita della versione 5.01 del Clipper e che presumibilmente permetterà una gestione del file eseguibile simile a quella del RTLink (i produttori del Blinker, prima di rilasciare questo upgrade, vogliono verificare sulla nuova versione del Clipper come sono stati risolti alcuni problemi relativi alla gestione della memoria evidenziati dall'ultima release del compilatore).

Ma le novità di questo linker non si esauriscono certo nella particolare modalità di gestione degli overlay e nella enorme velocità operativa; utilissima

per il programmatore si rivela infatti la «Blinker Programming Interface», una serie di funzioni realizzate con l'Extended System del Clipper, richiamabili dalle applicazioni realizzate tramite il Blinker, con le quali è possibile controllare l'esecuzione del programma oltre che avere informazioni utili per le fasi di «debug» (ricerca errori) e «profiling» (ottimizzazione del codice).

Tra queste funzioni, ho trovato di particolare interesse quelle che permettono di realizzare versioni dimostrative del proprio programma senza la necessità di dover apportare sostanziali modifiche al codice sorgente; tramite le funzioni BLIDEMDTE, BLIDEMMIN e BLIDEMCAL è possibile definire, rispettivamente, una data, un limite in minuti ed un limite in numero di chiamate a funzioni oltre al quale il programma cessa di funzionare.

Unitamente a queste funzioni è possibile sfruttare la BLISERNUM che permette di immergere nel file EXE una stringa di lunghezza massima pari a 50 caratteri in forma codificata, così da non poter essere individuata tramite i normali editor binari; in questo modo è possibile immergere nel corpo del programma informazioni riservate (quali, ad esempio, il nome del cliente a cui si è concessa la licenza di utilizzo del pacchetto o il numero di serie) che durante la normale esecuzione possono essere richiamate e decrittografate per controllo.

Altrettanto utili si sono rivelate le funzioni che permettono di controllare lo stato di frammentazione della memoria in seguito allo «swapping» degli overlay e quelle che permettono di effettuare una ricompattazione dello spazio di memoria ad intervalli regolari definibili da parte del programmatore.

Il «debug & profiling» dei programmi è poi facilitato da funzioni quali la BLIDISSTS (che riporta informazioni sugli overlay attualmente attivi in memoria), la BLIPRFMOD (che in unione al modulo BLPRFPRG.PRG fornito con il Blinker permette di crearsi delle procedure di ottimizzazione del codice personalizzate) e da una serie di funzioni che permettono di avere in ogni istante sotto controllo la situazione dei vari moduli componenti il programma.

In definitiva il Blinker oltre a svolgere la funzione specifica di collegamento di più moduli oggetto in un unico file eseguibile, fornisce degli strumenti estremamente potenti nelle mani del programmatore che vede ridursi in maniera consistente i tempi necessari allo sviluppo oltre che aumentare in maniera sensibile la qualità dei programmi prodotti, grazie alle funzioni di «profiling» messi a disposizione dal Blinker.

C' è un termine della lingua inglese che mi ha sempre colpito per la sua concisione e per l'immediatezza con cui è in grado di trasferire il concetto ad esso collegato: «definitive», che potremmo tradurre nella nostra lingua come «decisivo», «finale». Questo termine viene sovente utilizzato nel presentare la Funky Lib, una delle più complete librerie di funzioni per Clipper che siano mai state immesse sul mercato.

Realizzata da Dirk Lesko, che a seguito del successo ottenuto dal suo prodotto ha fondato la Lesko Inc., si propone come la «panacea» del programmatore Clipper per il gran numero di funzioni che comprende; confesso quindi di aver aperto con vivo interesse la confezione della Funky Lib (arrivata ora alla versione 1.50), curioso di verificare come questa libreria si integri con la nuova versione del Clipper, già di per sé ampliata notevolmente a livello di funzionalità e strumenti di programmazione messi a disposizione del programmatore.

La confezione è composta da un contenitore di cartone rigido al cui interno troviamo un raccoglitore estraibile ad anelli contenente le 370 pagine del manuale oltre ai floppy (6 nel formato da 5" 1/4) su cui troviamo un gran numero di file raggruppati, per specifici argomenti, in archivi autodecompattanti. La libreria è utilizzabile sia con la versione Summer 87 che con la più recente versione 5; per chi intendesse utilizzare il Blinker al posto del RTlink viene fornita una speciale libreria che permette di effettuare l'overlay dinamico delle funzioni C e Assembler della Funky Lib.

Curiosando tra gli altri file che troviamo sull'hard-disk dopo la decompattazione dei vari archivi, sono da segnalare:

- uno speciale .OBJ per gestire la visualizzazione a 43/50 linee su schermo VGA, da linkare al proprio applicativo se questi utilizza una delle funzioni per la gestione del video in modalità ad alta risoluzione; nella prossima versione della Funky Lib probabilmente questo modulo non sarà più necessario in quanto sarà disponibile una API (Application Program Interface) del Clipper migliorata;

- una utility per la creazione di «script file» per il linker RTlink per permettere la creazione di Pre-Link Library (PLL), caratteristica quest'ultima innovativa del Clipper 5;

- i moduli database di documentazione per le Norton Guide ed il Tom Retting Help, i due sistemi più diffusi tra i programmatori per la documentazione on-line;



Funky Lib

- due utility, dCode e dMenu, destinate rispettivamente alla generazione della documentazione tecnica del programma in fase di realizzazione ed alla gestione tramite menu e file batch dei programmi contenuti sul proprio hard disk;

- i sorgenti di alcune delle funzioni realizzate in Clipper oltre ad una serie di programmi demo che guidano il programmatore nell'utilizzo delle funzioni più complesse della Funky Lib.

Dalla corposità del manuale si intuisce come il numero delle funzioni presenti in questa libreria sia veramente elevato; si tratta infatti di oltre 450 funzioni suddivise in 9 categorie distinte. Vediamo ora, per ognuna di queste categorie, una carrellata sulle principali funzioni che vengono messe a disposizione del programmatore:

Array: già con il Clipper 5 la gestione degli array è stata notevolmente ampliata, con l'introduzione, tra le altre cose, degli array bidimensionali; le funzioni presenti nella Funky Lib permettono di manipolare facilmente questi oggetti, con comode routine per la ricerca e l'inserimento di valori, calcolo di minimi e massimi sia numerici che di lunghezza stringhe;

Disco: 25 funzioni che permettono al programmatore di avere il pieno controllo delle unità di memoria di massa del

PC, siano esse floppy o hard disk, con la possibilità di creare o cambiare directory, ottenere lo spazio libero su disco ed ottenere lo stato di un particolare dispositivo (fisso, rimovibile, ecc.); troviamo inoltre funzioni per la gestione a basso livello di queste unità (gestione del numero di cluster e settori) e la possibilità di monitorare se un determinato disco è accessibile o meno (ad esempio per verificare che sia stato o meno chiuso lo sportellino dell'unità floppy).

Environment: le 30 funzioni di questa sezione permettono di accedere ai parametri dell'ambiente di lavoro (environment) settati tramite i comandi SET «funzione», di effettuare controlli sulla data (anno bisestile, ultimo giorno di un determinato mese, ecc.) e conversioni tra un formato e l'altro di data (numerico o stringa), di generare numeri secondo sequenze random e di inserire comandi nel buffer di tastiera.

Mouse: questa è forse una delle più interessanti sezioni della Funky Lib. Ben 60 funzioni sono a disposizione del programmatore per avere il completo controllo del mouse nelle proprie applicazioni; si va dalle funzioni per la gestione a basso livello (cursore on/off, posizionamento per coordinate, coordinate massime e minime permesse al cursore, numero di bottoni del mouse, ecc.)

fino a macro-funzioni dall'estrema potenzialità che permettono di creare menu di scelta, «pick list» di file e directory oltre a sistemi di «browse» con il completo controllo tramite il mouse già al loro interno, permettendo un risparmio di tempo in fase di stesura di un applicativo veramente considerevole.

Stringhe: con le oltre 40 funzioni di questa sezione della Funky Lib la manipolazione di stringhe in un applicativo si semplifica notevolmente; le stringhe possono, infatti, essere convertite da ASCII in EBCDIC (e viceversa), criptate in maniera tale da non poter essere lette se non dopo opportuna decodifica, manipolate sia a livello di carattere che a livello di bit del singolo carattere, convertite in maiuscolo e minuscolo e scandite alla ricerca di particolari occorrenze.

Sistema: in questa sezione confluiscono tutte le funzioni della Funky Lib dedicate alla gestione dell'hardware presente sulla macchina. Troviamo quindi funzioni per la gestione delle porte seriale e parallela, per l'I/O su una qualsiasi delle porte nello spazio di indirizzamento del PC, per la verifica del tipo di CPU ed eventuale coprocessore matematico installati sulla macchina oltre a funzioni abbastanza particolari come la CRASH() che effettua un blocco totale del PC con conseguente reboot hardware (!). Sono disponibili inoltre, in questa sezione, funzioni per effettuare le tipiche operazioni binarie (and, or, not, ecc.), per la gestione degli interrupt (sia hardware che software), per la gestione della coda di stampa del DOS, per la consultazione di alcuni parametri del BIOS (data, tastiere supportate, ecc.), per conoscere la quantità di memoria disponibile (sia essa quella base che quella espansa o estesa), per la gestione di un orologio digitale ed addirittura per la generazione di suoni in puro stile «argade game»! Si tratta di un centinaio di funzioni che permettono di utilizzare il Clipper non solo per la realizzazione di sistemi di gestione di basi di dati, ma anche per la realizzazione di applicativi nei settori più disparati, eliminando ogni limite intrinseco di questo linguaggio di programmazione.

Video: un altro cavallo di battaglia della Funky Lib è la sezione relativa alla gestione del video; con le funzioni in essa contenute è possibile arricchire i nostri applicativi con tutte le capacità grafiche offerte dall'alta risoluzione delle schede VGA, sia in modalità grafica che testo. Oltre 40 funzioni con le quali è possibile conoscere il tipo di adattatore video installato sul PC (e rendere così parametrico ed autoconfigurante il proprio programma), disegnare box e window sullo schermo, lavorare in mo-



Schermate della Funky. Due esempi di «pick-list», font e modalità grafiche disponibili.



dalità testo VGA a 43 o 50 linee, sfruttare fino a 64 colori su schede EGA e VGA, salvare e recuperare da disco parti di schermo, commutare da una modalità di visualizzazione ad un'altra ed effettuare degli «scroll» del video (anche solo una parte) nelle quattro direzioni.

Miscellaneous: sotto questa voce vengono raccolte tutte quelle funzioni non riconducibili ad una specifica tipologia; tra queste troviamo funzioni per il caricamento di nuovi font di caratteri nella memoria delle schede EGA e VGA, per il reboot («caldo» o «freddo») del PC, per vincolare l'accesso ai floppy disk sotto determinate condizioni, per attivare e disattivare la tastiera e per parcheggiare le testine dell'hard disk sull'ultimo settore. Oltre a queste funzioni, già di per sé molto potenti, ne troviamo altre un po' particolari come la TIMEOUT() che permette di specificare una funzione da eseguire durante gli stati di attesa o la NAP() che permette di sospendere l'esecuzione del programma per un numero prefissato di cicli di clock di sistema.

Che dire, è una vera ubriacatura di funzioni! Vi assicuro che mentre sfogliavo il manuale della Funky Lib più volte mi è capitato di pensare: «Se avessi

avuto questa funzione quella volta che...».

Un grande vantaggio della Funky Lib è poi la sua integrazione con altri pacchetti per Clipper; infatti un fattore molto importante da tenere in considerazione quando si decide l'acquisto di una libreria per un determinato compilatore è la sua facilità d'integrazione con altri tool di sviluppo. Nel caso della Funky Lib, data la sua notorietà, questo aspetto è estremamente semplificato in quanto la maggior parte di prodotti per Clipper (come, ad esempio, il Blinker e la DGE oggetti di questa prova) contengono al loro interno le informazioni relative ad ottimizzare l'uso di questa libreria in congiunzione con i loro tool, sia in fase di stesura dell'applicazione che in fase di link per evitare eventuali conflitti con altre dichiarazioni di funzioni.

Se siete intenzionati ad affidarvi ad un'unica libreria da cui attingere nuove funzionalità per i vostri programmi, la Funky Lib è probabilmente quello che state cercando, un ottimo investimento che vi assicura sempre una fonte inesauribile di strumenti di lavoro per rendere più facile il vostro lavoro di programmazione e più professionali i vostri applicativi.

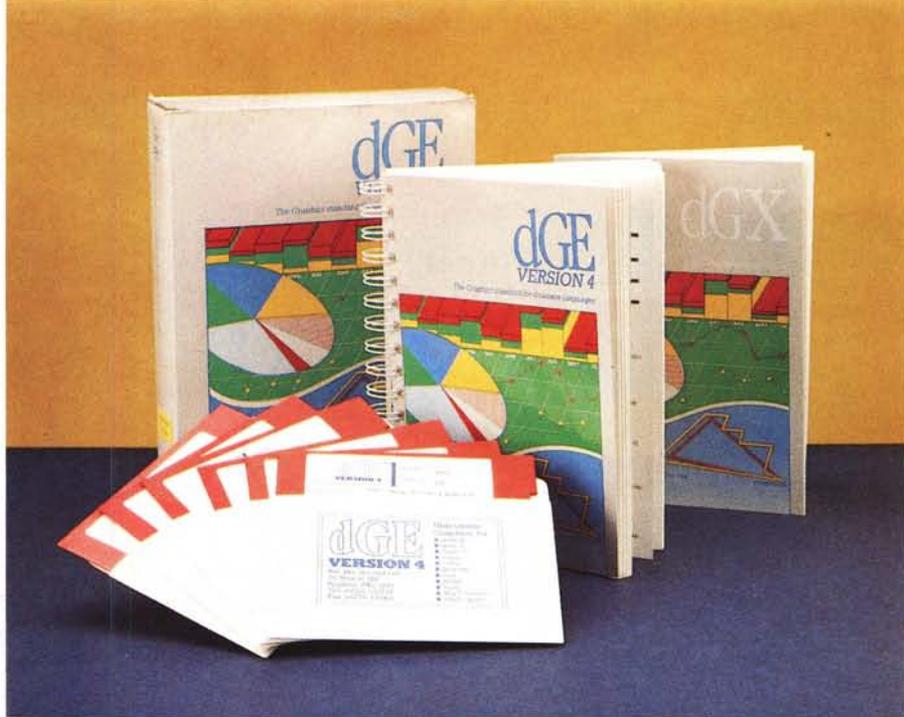
Con l'avvento di nuovi standard grafici ad alta risoluzione, il «look» dei programmi in ambiente MS-DOS ha subito un rapido miglioramento ed ormai ci si sta avviando verso una standardizzazione delle interfacce grafiche; ogni programma DOS che voglia essere almeno un po' «accattivante» deve permettere un utilizzo semplice ed intuitivo delle sue funzionalità, con menu che si «srotolano» dall'alto dello schermo alla pressione di un tasto o al «click» di un mouse.

Sovente, inoltre, si ha la necessità di presentare in forma grafica i risultati di determinate elaborazioni o di arricchire con grafici esemplificativi la produzione di report di stampa, soprattutto in campi, quali quelli della gestione di basi di dati, fertile terreno per i programmatori Clipper.

Tra i tool disponibili per questo linguaggio è stata ora rilasciata la versione 4 della DGE, libreria grafica dall'uso generale, nel senso che oltre ad essere interfacciabile con l'ultima versione del Clipper può essere utilizzata per arricchire le funzionalità di programmi scritti con altri compilatori dBASE o C, riducendo così l'investimento monetario necessario ad un programmatore per dotarsi di uno strumento grafico ed uniformando il suo stile programmatico oltre che l'aspetto esteriore che potrebbe dare un «imprimatur» unico ai suoi prodotti.

Tramite questa libreria è possibile estendere il set di funzioni del proprio linguaggio di programmazione con comandi per la creazione di grafici ad alta risoluzione, sia a 2 che a 3 dimensioni, che possono essere sia scambiati con altri applicativi che stampati sui più comuni dispositivi di stampa in ambiente MS-DOS; unitamente alla libreria vengono forniti due tool di ausilio al programmatore nella realizzazione dei grafici e di nuovi font di caratteri per il testo (oltre a quelli già in dotazione) che eventualmente vi dovesse essere inserito.

L'installazione su disco fisso della libreria avviene tramite un apposito programma, contenuto sul primo dei 6 floppy da 5" 1/4 contenuti nella confezione, che si occupa di copiare nella directory prescelta dall'utente i file relativi all'adattatore video con il quale il programma che si intende realizzare deve operare (è possibile comunque installarli tutti e cancellare in un secondo momento quelli che non interessano); analogamente viene richiesto all'utente di indicare quale linguaggio di programmazione intende utilizzare con la DGE, in modo che vengano copiati solo i moduli ad esso relativo (il manuale consiglia infatti di installare la libreria ed i suoi file



DGE Library

accessori in directory differenti a seconda del linguaggio in modo da non mischiare tra di loro i diversi moduli). Le Norton Guide (per una consultazione on-line della manualistica tramite il famoso programma della Norton Computing), i file con i font predefiniti e le icone, alcune utility e programmi demo completano la lista di materiale che troviamo sul nostro hard disk al termine dell'installazione.

Come consigliato dal manuale la prima operazione che ho fatto al termine dell'installazione è stata quella di compilare con il Clipper 5 il programma dimostrativo fornito con la libreria e... ammirare! Tramite un menu di scelta è infatti possibile rendersi subito conto delle enormi potenzialità di questa libreria: grafici nelle più svariate tipologie, disegni arricchiti da font di tutte le dimensioni, addirittura un oscilloscopio con tanto di display su cui vengono visualizzate in movimento in tempo reale due diverse forme d'onda.

Si rimane veramente sbalorditi dalle potenzialità di questa libreria e non si fatica molto a comprendere il perché sia da molti anni considerata uno dei migliori tool disponibili sul mercato per i programmatori Clipper (e non solo); ma dopo questa ubriacatura di colori del demo passiamo subito ad analizzare nel dettaglio le funzionalità offerte da questo pacchetto, vedendo innanzitutto quali sono i passi necessari per richiamare le funzioni dal proprio applicativo.

Il controllo delle modalità grafiche della scheda video prescelta avviene trami-

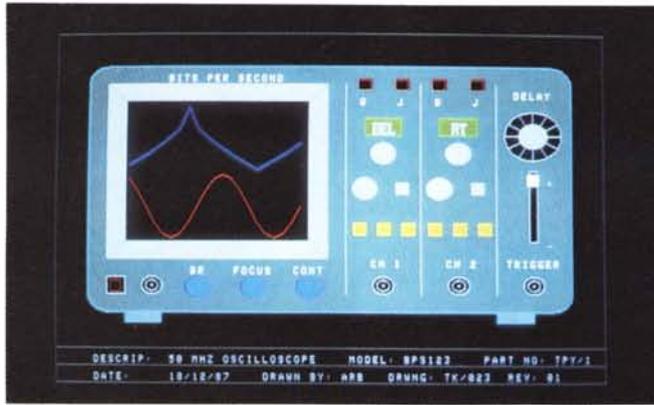
te un specifico modulo Assembler che può essere caricato residente in memoria (TSR) o linkato direttamente alla propria applicazione; è chiaro che la prima possibilità, avendo lo svantaggio di occupare preziosa RAM (a meno di non sfruttare la memoria espansa) è consigliabile soltanto con linguaggi interpretati (quali il dBASE) mentre in Clipper o in C sarà più conveniente collegare questo modulo direttamente al modulo .EXE del programma prodotto (l'incremento in termini di spazio occupato in memoria è di 50 Kbyte).

Per ogni adattatore video (CGA, EGA, VGA, ecc..) esiste uno specifico modulo di interfaccia, le cui funzionalità sono richiamabili da programma con istruzioni del tipo:

```
Result = FunctionName(Param-1,...,ParamN).
```

Le funzioni presenti in questa libreria coprono, come già detto in precedenza, un vasto ventaglio di tipologie, sia in modalità raster che vettoriale (anche per quanto riguarda i font per il testo); i grafici possono essere sovrapposti tra di loro e grafici di tipo differente possono essere combinati in un'unica schermata. Tra i principali tipi di grafici disponibili troviamo i classici Bar, Pie, Polar, Polyline, High-Low-Close oltre ad alcuni più particolari quali Gantt e Time-Series (quest'ultimo utilissimo per la visualizzazione di serie continue di dati provenienti da un input esterno, come nell'esempio dell'oscilloscopio visto a proposito del demo).

È chiaramente supportato l'uso del



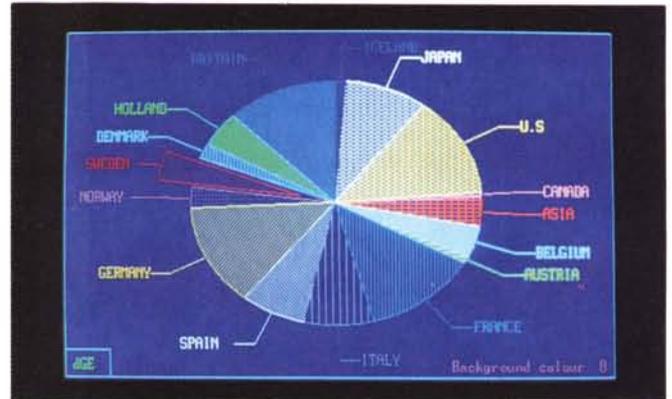
mouse, in tutte le modalità grafiche, tramite una dozzina di funzioni che svincolano il programmatore da strani giochi con gli interrupt (peraltro non facilissimi in Clipper); se a queste funzioni associamo quelle per il disegno di linee, archi e cerchi, ugualmente potenti e complete, abbiamo a disposizione tutti gli strumenti necessari per realizzare un modulo di disegno grafico all'interno del nostro applicativo.

I grafici e le immagini create con la DGE possono inoltre essere scambiate con programmi di Word Processing o DTP grazie al fatto che questa libreria, tramite la funzione PCLWRITE(), permette la creazione di file nel formato PCX, ormai uno degli standard più diffusi nel mondo della grafica per computer; analogamente, con le funzioni PRINTFILE e PLOTON è possibile creare in uscita dei file nel formato vettoriale HP-GL (standard Hewlett Packard nel campo dei dispositivi di plottaggio). La differenza tra i due formati consiste sostanzialmente nel fatto che il secondo permette una più facile scalatura delle immagini oltre ad una maggiore indipendenza dall'adattatore video.

La stampa delle immagini e dei grafici creati tramite le funzioni di questa libreria può avvenire su diversi tipi di dispositivi. Innanzitutto è possibile redirigere l'output su un file, per successive elaborazioni o stampe off-line; per l'output su stampanti a matrice è invece richiesta la compatibilità del dispositivo con lo standard Epson (codici grafici) ed avere una testa di stampa con almeno 8 aghi, mentre per stampanti laser, quale la HP Laserjet, è disponibile l'output nel formato PCL (Printer Command Language) o PostScript.

Diamo ora un'occhiata ai due tool forniti in dotazione a cui abbiamo fatto cenno all'inizio della prova; il primo di questi è il dGX (Graphics Design Centre), un ambiente di sviluppo interattivo che permette, a partire da un file di dati,

Schermate della DGE. Alcuni esempi tratti dal demo di questo pacchetto.



di scegliere i valori di uno o più campi ed il tipo di grafico con cui visualizzarli per poter avere un «preview» del risultato: se quest'ultimo è ritenuto soddisfacente da parte del programmatore, viene generato direttamente da quest'ambiente la parte di codice da inserire nel programma finale per riottenere il medesimo grafico da quella combinazione di dati.

Questo tool, la cui documentazione è contenuta in un manuale separato rispetto a quello della DGE, si rivela di estrema utilità nella fase di sviluppo di un applicativo in quanto permette di ridurre in maniera considerevole i tempi di messa a punto delle modalità di visualizzazione dei dati, producendo in maniera diretta il codice bello e pronto e permettendo di modificare in continuazione l'immagine fino ad ottenere il risultato più soddisfacente senza dover ogni volta ricompilare il programma.

Il secondo è invece un editor di font (GFont) che permette al programmatore di creare dei set di caratteri dal «look» personalizzato, caratteri che possono essere definiti con una matrice di dimensione sia fissa che variabile (font Supertext); oltre che per la definizione di nuovi font, questa utility è di ausilio nella creazione di «icone», fino ad un massimo di 8, ciascuna dalle dimensioni di 64*64 punti.

La DGE si presenta, in definitiva,

completa di tutti gli elementi necessari a realizzare applicazioni per la gestione di basi di dati (e non solo) dall'aspetto grafico molto professionale, che vi permetterà di dare un «tocco in più» alle vostre applicazioni.

Conclusione

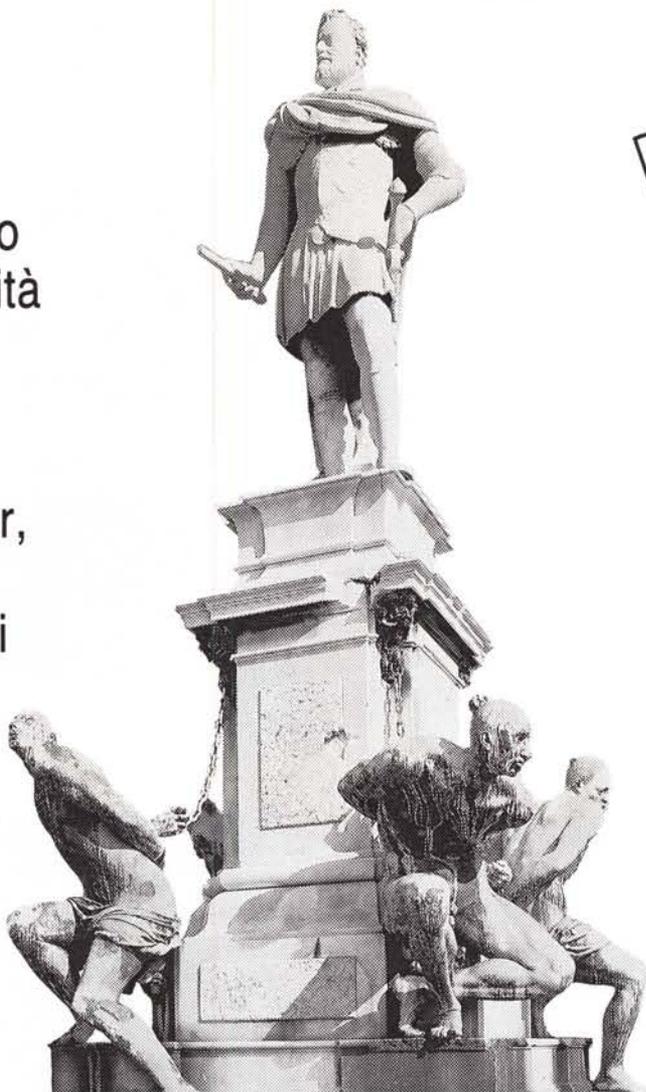
I tre pacchetti oggetto di questa prova sono considerati tra i migliori tool per gli sviluppatori in ambiente Clipper ed alla prova dei fatti lo hanno dimostrato ampiamente ma... si sa... la qualità si paga! Ognuno di questi tool ha un prezzo che si aggira intorno alle 700 mila lire, quindi più alla portata del programmatore professionista, che può ammortizzare facilmente la spesa, che dell'hobbysta; il loro prezzo è comunque pienamente giustificato dalla completezza di funzionalità di cui sono dotati e dall'enorme risparmio di tempo che garantiscono durante la stesura di un applicativo, soprattutto se questo è di grosse dimensioni.

Allora, se non siete completamente soddisfatti di come procede la realizzazione del vostro ultimo programma o se non vi va di perdere troppo tempo a scrivervi da soli «quella tal funzione che...», vi consiglio vivamente di dare un'occhiata ad uno di questi pacchetti: sono sicuro che troverete qualcosa che fa al caso vostro!

MB

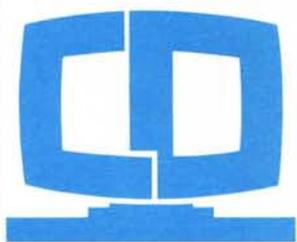
A Livorno cresce il successo del Computer

La catena italiana dell'informatica è arrivata a Livorno e presenta la qualità della sua gamma ad un prezzo conveniente. Personal Computer, accessori e novità vi saranno illustrati da personale qualificato. Assistenza diretta.



LIVORNO
Via Garibaldi, 112
Tel. 0586/210230

È gradita la Vs. visita
per il ritiro gratuito del nuovo
Catalogo Generale



COMPUTER DISCOUNT

la catena italiana dell'informatica

CARRY-I

LA FAMIGLIA IN ESPANSIONE



SERIE 8000
PERSONAL COMPUTER XT AT HD FINO 40MB DISPLAY DUAL

SERIE 9000
PERSONAL COMPUTER AT-386SX HD FINO 80MB DISPLAY VGA

SERIE 6000
STAZIONI DI RETE XT AT (ETHERNET-ARCNET)

Importatore per l'Italia:
PRIMA COMPUTER TRADING - REGGIO EMILIA
Distributore per l'Italia:
Andromeda S.R.L. - Via Umbria 16 - 42100 REGGIO EMILIA
Tel. 0522-511212/511201 - Fax 0522-513404