

Spediz. in abbonamento postale GR-11770 L. 2.200
(...)

49 CORSO PRATICO COL COMPUTER

422014

è una iniziativa
FABBRI EDITORI

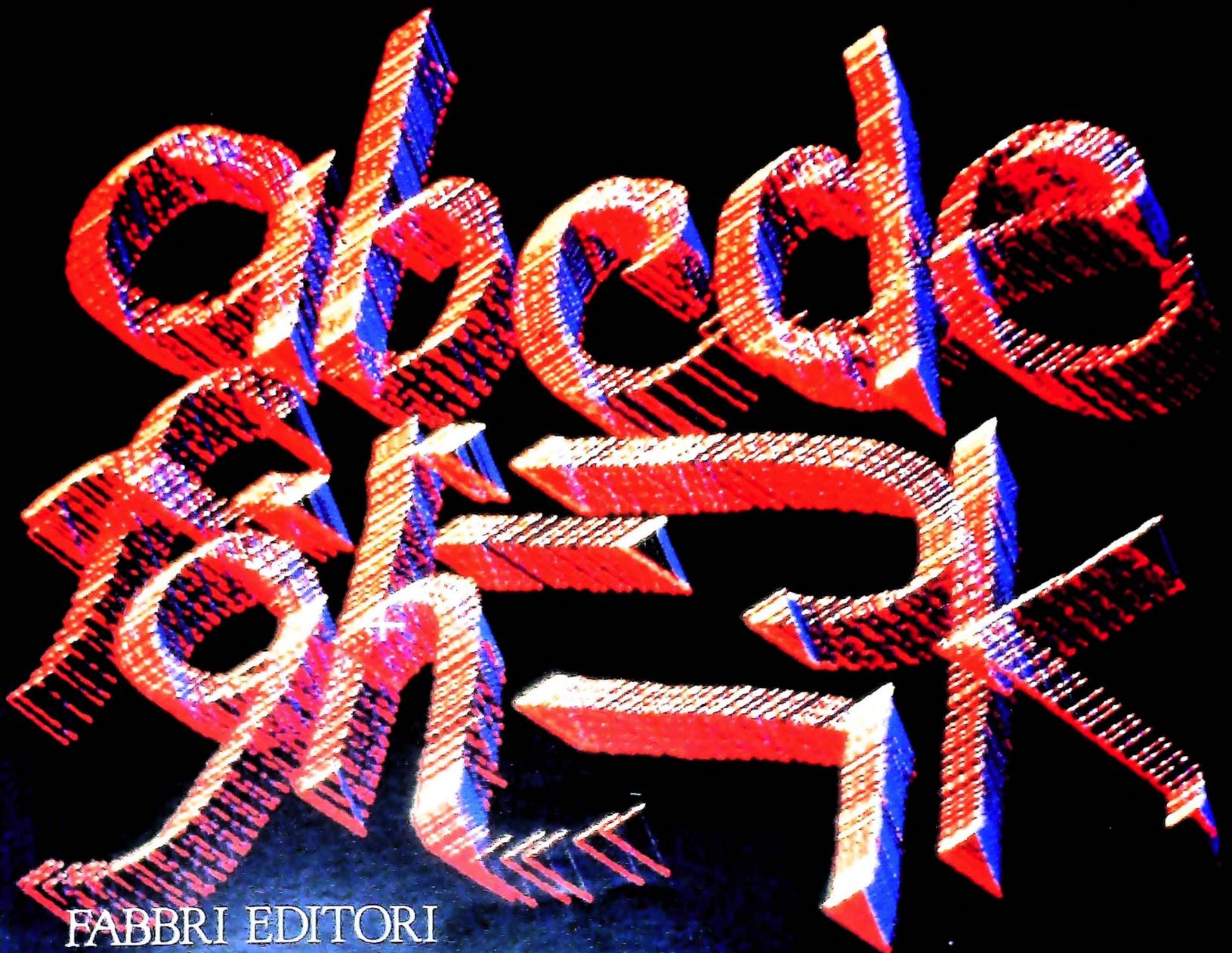
in collaborazione con
BANCO DI ROMA

e **OLIVETTI**

F4 F5 F6 F7 F8

diretto da **GIANNI DIGLI ANTONI**

BATTERY LOW



FABBRI EDITORI

IL BANCO DI ROMA FINANZIA IL VOSTRO ACQUISTO DI M 10 e M 20

Acquisto per contanti

È la formula di acquisto tradizionale.

Non vi sono particolari commenti da fare, se non sottolineare che troverete ampia disponibilità presso i punti di vendita Olivetti, poiché, grazie al "Corso pratico col computer", godrete di un rapporto di privilegio.

Il servizio di finanziamento bancario

Le seguenti norme descrivono dettagliatamente il servizio di finanziamento offerto dal Banco di Roma e dagli Istituti bancari a esso collegati:

Banca Centro Sud
Banco di Perugia

Le agenzie e/o sportelli di questi istituti sono presenti in 216 località italiane.

Come si accede al credito e come si entra in possesso del computer

- 1) Il Banco di Roma produce una modulistica che è stata distribuita a tutti i punti di vendita dei computer M 10 e M 20 caratterizzati dalla vetrofanìa M 10.
- 2) L'accesso al servizio bancario è limitato solo a coloro che si presenteranno al punto di vendita Olivetti.
- 3) Il punto di vendita Olivetti provvederà a istruire la pratica con la più vicina agenzia del Banco di Roma, a comunicare al cliente entro pochi giorni l'avvenuta concessione del credito e a consegnare il computer.

I valori del credito

Le convenzioni messe a punto con il Banco di Roma, valide anche per le banche collegate, prevedono:

- 1) Il credito non ha un limite minimo, purché tra le parti acquistate vi sia l'unità computer base.
- 2) Il valore massimo unitario per il credito è fissato nei seguenti termini:
 - valore massimo unitario per M 10 = L. 3.000.000
 - valore massimo unitario per M 20 = L. 15.000.000
- 3) Il tasso passivo applicato al cliente è pari

al "prime rate ABI (Associazione Bancaria Italiana) + 1,5 punti percentuali".

- 4) La convenzione prevede anche l'adeguamento del tasso passivo applicato al cliente a ogni variazione del "prime rate ABI"; tale adeguamento avverrà fin dal mese successivo a quello a cui è avvenuta la variazione.
- 5) La capitalizzazione degli interessi è annuale con rate di rimborso costanti, mensili, posticipate; il periodo del prestito è fissato in 18 mesi.
- 6) Al cliente è richiesto, a titolo di impegno, un deposito cauzionale pari al 10% del valore del prodotto acquistato, IVA inclusa; di tale 10% L. 50.000 saranno trattene dal Banco di Roma a titolo di rimborso spese per l'istruttoria, il rimanente valore sarà vincolato come deposito fruttifero a un tasso annuo pari all'11%, per tutta la durata del prestito e verrà utilizzato quale rimborso delle ultime rate.
- 7) Nel caso in cui il cliente acquisti in un momento successivo altre parti del computer (esempio, stampante) con la formula del finanziamento bancario, tale nuovo prestito attiverà un nuovo contratto con gli stessi termini temporali e finanziari del precedente.

Le diverse forme di pagamento del finanziamento bancario

Il pagamento potrà avvenire:

- presso l'agenzia del Banco di Roma, o Istituti bancari a esso collegati, più vicina al punto di vendita Olivetti;
- presso qualsiasi altra agenzia del Banco di Roma, o Istituto a esso collegati;
- presso qualsiasi sportello di qualsiasi Istituto bancario, tramite ordine di bonifico (che potrà essere fatto una volta e avrà valore per tutte le rate);
- presso qualsiasi Ufficio Postale, tramite vaglia o conto corrente postale. Il numero di conto corrente postale sul quale effettuare il versamento verrà fornito dall'agenzia del Banco di Roma, o da Istituti a esso collegati.

 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.

Direttore dell'opera
GIANNI DEGLI ANTONI

Comitato Scientifico
GIANNI DEGLI ANTONI
Docente di Teoria dell'Informazione, Direttore dell'Istituto di Cibernetica dell'Università degli Studi di Milano

UMBERTO ECO
Ordinario di Semiotica presso l'Università di Bologna

MARIO ITALIANI
Ordinario di Teoria e Applicazione delle Macchine Calcolatrici presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

MARCO MAIOCCHI
Professore Incaricato di Teoria e Applicazione delle Macchine Calcolatrici presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

DANIELE MARINI
Ricerca universitaria presso l'Istituto di Cibernetica dell'Università degli Studi di Milano

Curatori di rubriche
MARCO ANELLI, DIEGO BIASI, ANDREA GRANELLI, ALDO GRASSO, MARCO MAIOCCHI, DANIELE MARINI, GIANCARLO MAURI, CLAUDIO PARMELLI

Testi
MARCO ANELLI, DIEGO BIASI, CLAUDIO MORDÀ, Eidos (TIZIANO BRUGNETTI), Etnoteam (ADRIANA BICEGO)

Tavole
Logical Studio Communication
Il Corso di Programmazione e BASIC è stato realizzato da Etnoteam S.p.A., Milano
Computergrafica è stato realizzato da Eidos, S.c.r.l., Milano
Usare il Computer è stato realizzato in collaborazione con PARSEC S.N.C. - Milano

Direttore Editoriale
ORSOLA FENGLI

Redazione
CARLA VERGANI
LOGICAL STUDIO COMMUNICATION

Art Director
CESARE BARONI

Impaginazione
BRUNO DE CHECCHI
PAOLA ROZZA

Programmazione Editoriale
ROSANNA ZERBARINI
GIOVANNA BREGGÈ

Segretaria di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

Corso Pratico col Computer - Copyright © sul fascicolo 1985 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A., Milano - Copyright © sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Etas S.p.A., Milano - Prima Edizione 1984 - Direttore responsabile GIOVANNI GIOVANNINI - Registrazione presso il Tribunale di Milano n. 135 del 10 marzo 1984 - Iscrizione al Registro Nazionale della Stampa n. 00262, vol. 3, Foglio 489 del 20.9.1982 - Stampato presso lo Stabilimento Grafico del Gruppo Editoriale Fabbri S.p.A., Milano - Diffusione - Distribuzione per l'Italia Gruppo Editoriale Fabbri S.p.A., via Mecenate, 91 - Milano - tel. 02/50951 - Pubblicazione periodica settimanale - Anno II - n. 49 - esce il giovedì - Spedizione in abb. postale - Gruppo II/70. L'Editore si riserva la facoltà di modificare il prezzo nel corso della pubblicazione, se costretto da mutate condizioni di mercato.

LA CREAZIONE DI UN PROGRAMMA

Tre sono le fasi principali: l'analisi del problema da risolvere, la progettazione e la stesura vera e propria.

Linguaggio macchina e linguaggio simbolico

La fase preliminare nella costruzione di un programma consiste nel descrivere l'algoritmo e i dati che risolvono il problema specifico in una forma simbolica propria del linguaggio di programmazione usato. Questo programma "sorgente" costituisce l'ingresso al programma traduttore associato al linguaggio, il cui obiettivo è quello di generare in corrispondenza delle frasi in linguaggio sorgente le equivalenti frasi immediatamente interpretabili dalla macchina.

Infatti una macchina normalmente interpreta solamente il proprio linguaggio macchina o base, fatto di istruzioni elementari operanti su dati semplici, entrambi memorizzati nella memoria centrale come successione di bit la cui interpretazione è guidata dalla logica della macchina. Perché quindi un programma sia eseguibile deve essere nella forma accettabile dalla macchina; in particolare all'atto dell'esecuzione istruzioni e dati vengono individuati tramite il loro indirizzo assoluto di memoria che dovrà essere stato generato secondo modalità che vedremo di seguito.

Il linguaggio simbolico della macchina (assembly) è la forma più semplice di linguaggio di programmazione. Esso conserva la stessa struttura del linguaggio macchina, ma consente di specificare dati, indirizzi e istruzioni mediante nomi simbolici, sollevando così il programmatore dall'oneroso e inaffidabile compito di operare direttamente con la codificazione interna della macchina.

Il programma traduttore associato, chiamato *assemblatore*, esegue principalmente una traduzione uno a uno, cioè genera di massima una istruzione in linguaggio macchina per ogni frase del linguaggio simbolico e il suo compito più importante è quello di gestire gli indirizzi di dati e istruzioni assegnando locazioni assolute ai nomi simbolici, o costruendo le informazioni che verranno usate all'atto dell'esecuzione per determinare gli indirizzi assoluti di memoria.

Caricatore assoluto e rilocante

Una volta che l'assemblatore produce un *programma oggetto*, questo deve essere piazzato in memoria ed eseguito. Una immediata soluzione è che lo stesso assemblatore, dopo la tra-

duzione, muova l'oggetto in memoria e trasferisca il controllo a esso mediante una semplice istruzione di salto.

Tuttavia questo semplice schema, chiamato *assembly-and-go*, ha due evidenti svantaggi: spreco di memoria centrale, per la presenza contemporanea dell'assemblatore e del programma oggetto e necessità della traduzione ogni volta che viene eseguito il programma.

Per far fronte soprattutto al secondo inconveniente queste due fasi sono normalmente separate. L'assemblatore memorizza il programma oggetto su un dispositivo di memoria secondaria. Quindi successivamente un altro programma del sistema, il caricatore assoluto, carica in memoria centrale l'oggetto a partire da un indirizzo fissato e trasferisce il controllo a quell'indirizzo (figura pagina successiva, in alto).

In questo schema di caricamento, il caricatore esegue le seguenti due semplici funzioni:

a) *allocazione*

alloca lo spazio in memoria centrale al programma oggetto

b) *caricamento*

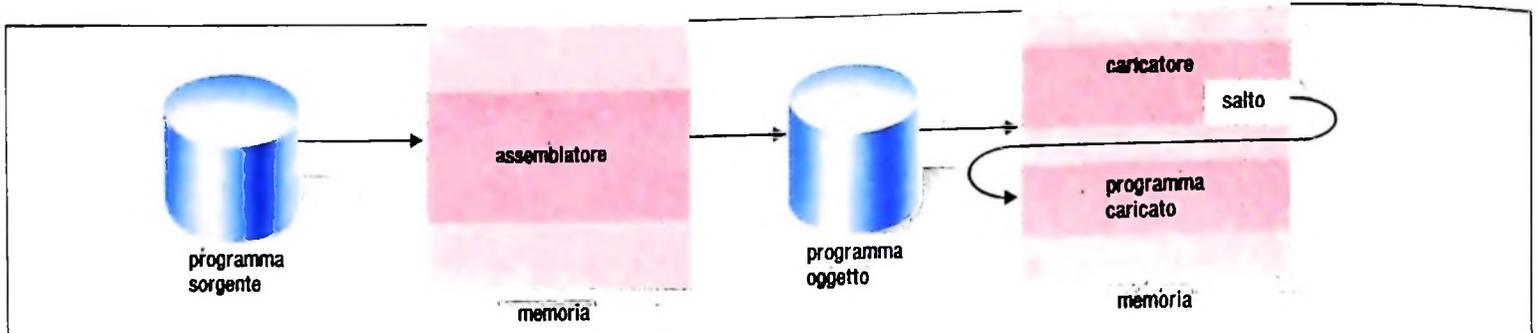
carica fisicamente le istruzioni e i dati del programma oggetto a partire dall'indirizzo di caricamento.

In particolare nel caso in cui tutti gli indirizzi nel programma oggetto siano assoluti, cioè già fissati in fase di assemblaggio, la sola informazione sull'indirizzo di caricamento del programma (specificata mediante una direttiva nel simbolico) è resa nota dall'assemblatore al caricatore.

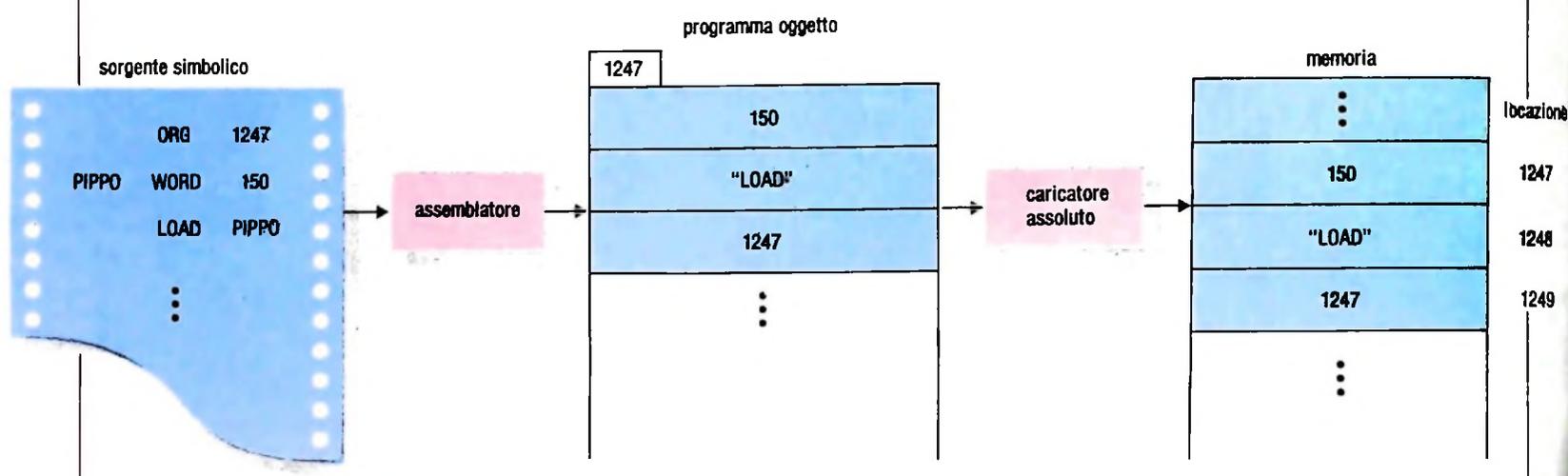
Una volta caricato nella giusta posizione di memoria e ricevuto il controllo, il programma eseguirà correttamente.

Nell'esempio riportato nella figura della pagina seguente, in basso, la direttiva ORG (origine) indica all'assemblatore che quanto segue deve essere caricato a partire dall'indirizzo assoluto 1247: in particolare una parola di memoria con nome simbolico PIPPO e contenuto 150 decimale, seguita da una istruzione di trasferimento del dato contenuto nella locazione PIPPO nel registro accumulatore della macchina. Evidentemente l'istruzione specificata opererà correttamente solo se il dato 150 è stato precedentemente caricato nella locazione di indirizzo assoluto 1247.

La grossa limitazione di questa soluzione è che il programmatore deve specificare in anticipo la posizione assoluta del programma. È importante quindi rendere parametrico il programma rispetto alla locazione fisica di installazione. A que-



In alto: fasi di assemblaggio e caricamento del programma oggetto. In basso: programma trasformato nella forma assoluta.



sto proposito l'assemblatore può generare un programma oggetto in cui gli indirizzi sono relativi a un indirizzo di caricamento non ancora definito. Per far questo l'assemblatore piazza le istruzioni e i dati in locazioni di memoria consecutive partendo dall'indirizzo 0.

Successivamente il caricatore corregge, ovvero riloca, tali riferimenti sommando un valore costante, l'indirizzo di caricamento, a ogni configurazione di bit che rappresenta un indirizzo relativo. Dopo questa fase di rilocazione il programma oggetto è stato trasformato nella forma assoluta e può a questo punto essere caricato fisicamente a partire dall'indirizzo di caricamento scelto.

Per la presenza di questa nuova funzione un tale caricatore è detto *rilocante*, mentre il programma oggetto è in una forma *rilocabile*. Questa flessibilità operativa ha tuttavia un costo aggiuntivo alla rilocazione. Poiché il caricatore non può fare alcuna ipotesi sulla interpretazione delle sequenze di bit formanti il programma rilocabile, l'assemblatore deve indicare quali parti di esso rappresentano indirizzi di memoria.

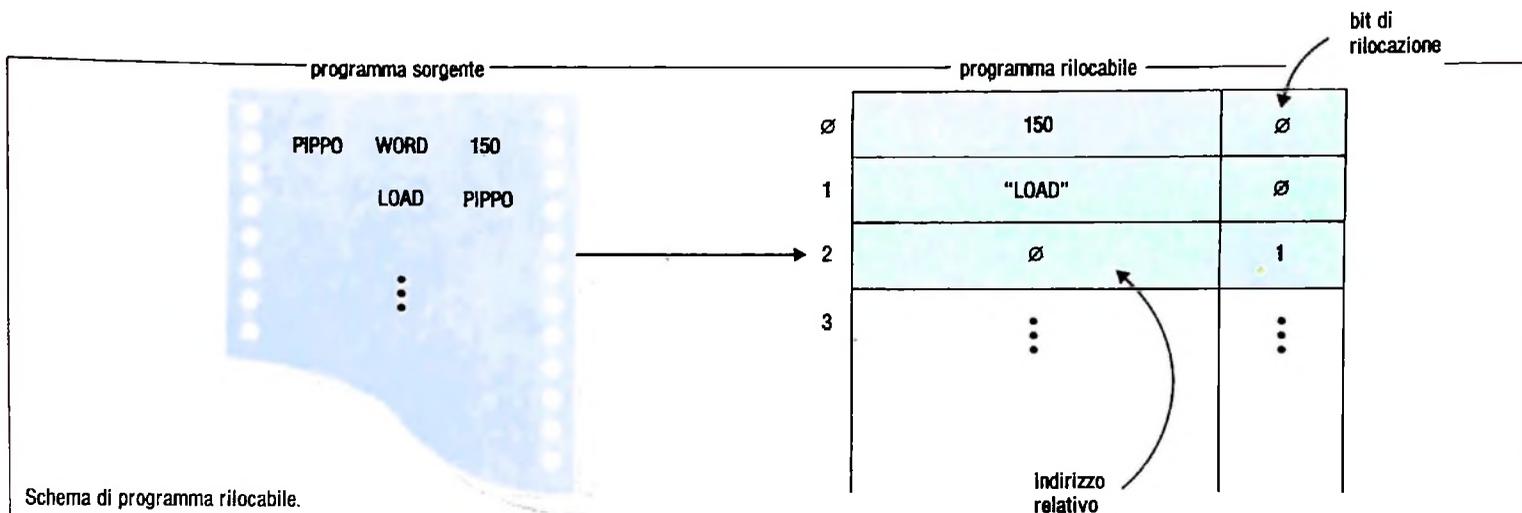
Supponendo per semplicità che gli indirizzi siano contenuti in una singola parola di memoria (così come il codice operativo della istruzione), il programma oggetto sarà formato dalle parole-contenuto (istruzioni e dati) più un bit di riloca-

zione per ogni parola (pagina seguente, in alto). Ovviamente l'informazione aggiuntiva sulla rilocazione serve solamente al caricatore e non compare più quando il programma è caricato.

Il problema della rilocazione può essere risolto anche direttamente dalla macchina. Per varie ragioni, molte macchine esplicitano l'indirizzo nelle istruzioni con una coppia di valori: il primo valore specifica il numero di un registro base, mentre il secondo un indirizzo relativo chiamato spiazzamento ("offset").

In fase di esecuzione la macchina risolve tale espressione sommando il valore contenuto nel registro base specificato con il valore dello spiazzamento, ottenendo un indirizzo assoluto di memoria.

Il risultato è che dovunque caricato in memoria tale programma funziona correttamente, se il registro base contiene l'indirizzo di caricamento. Il vantaggio è che nessun indirizzo necessita più della rilocazione statica fatta dal caricatore; il programma oggetto è direttamente caricabile in memoria ed eseguibile. In realtà tale registro base deve essere opportunamente caricato con l'indirizzo assoluto di caricamento; di conseguenza alcuni indirizzi nel programma oggetto dovranno ancora essere rilocati dal caricatore.



Collegatore

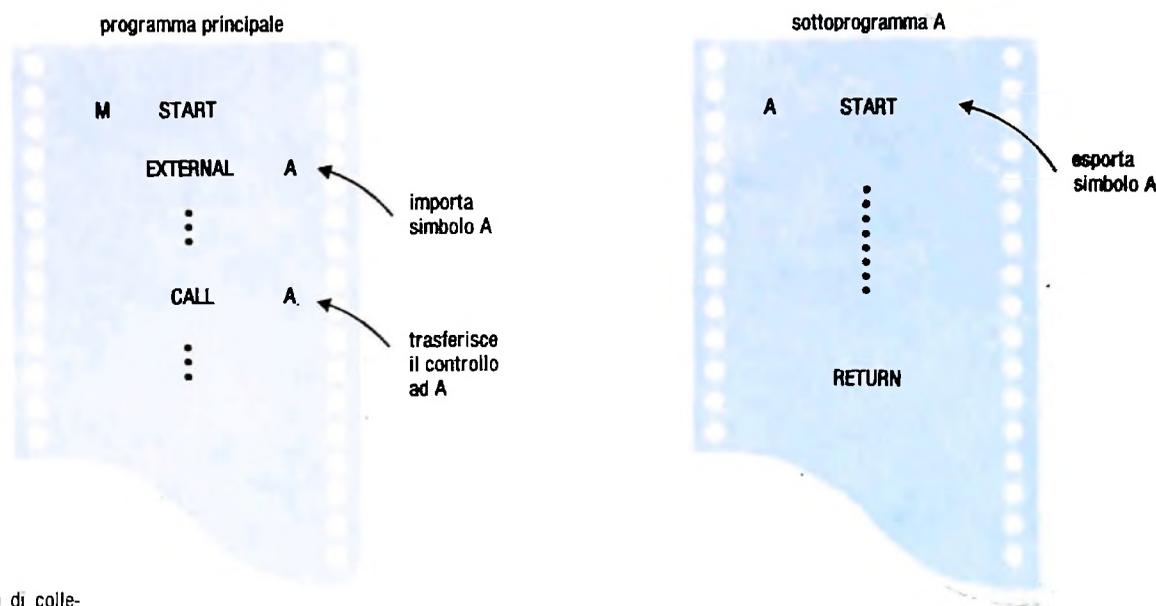
Una sentita esigenza nello sviluppo di programmi di grosse dimensioni è quella di poter suddividere il programma in moduli più piccoli, chiamati spesso sottoprogrammi. In questa situazione i moduli componenti possono essere sviluppati contemporaneamente da diversi programmatori, tradotti separatamente e quindi solo all'ultimo istante uniti insieme per formare il programma finale. Questa soluzione permette anche la costruzione di una libreria di sottoprogrammi ("moduli") riusabili e una facilità di manutenzione in quanto una modifica a un modulo necessita della sua sola traduzione. La funzione di collegare insieme tutti i moduli componenti e costruire il programma finale è compito di un altro programma del sistema, chiamato collegatore (o "linker").

Normalmente un programma utente è costituito da un programma principale e da un numero di sottoprogrammi tra-

dotti separatamente. Supponiamo di assemblare un programma principale M che ha nel suo codice una chiamata a un sottoprogramma A (figura in basso). Poiché A non è una locazione del programma M, l'assemblatore non è in grado di sostituire il nome simbolico con un indirizzo, non avendo alcuna informazione su di esso.

Possiamo infatti ipotizzare che tale sottoprogramma non sia stato ancora tradotto o addirittura codificato. La dichiarativa del linguaggio simbolico EXTERNAL A specifica proprio che A è un nome di dato o istruzione esterno al modulo. L'informazione sul simbolo esterno è opportunamente codificata dall'assemblatore e inserita nel programma oggetto.

Evidentemente perché possa essere risolto il riferimento, tale nome deve essere definito da un altro modulo. La dichiarativa di inizio sottoprogramma START A esporta infatti il nome simbolico all'esterno del modulo e una opportuna codifica di tale informazione è memorizzata nel relativo program-

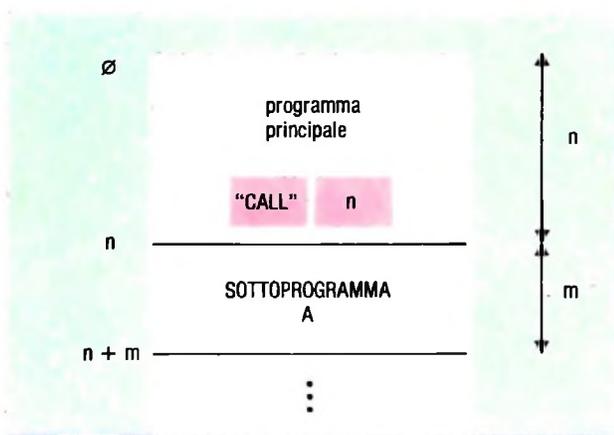


Ecco qui un esempio di collegamento tra moduli tradotti separatamente.

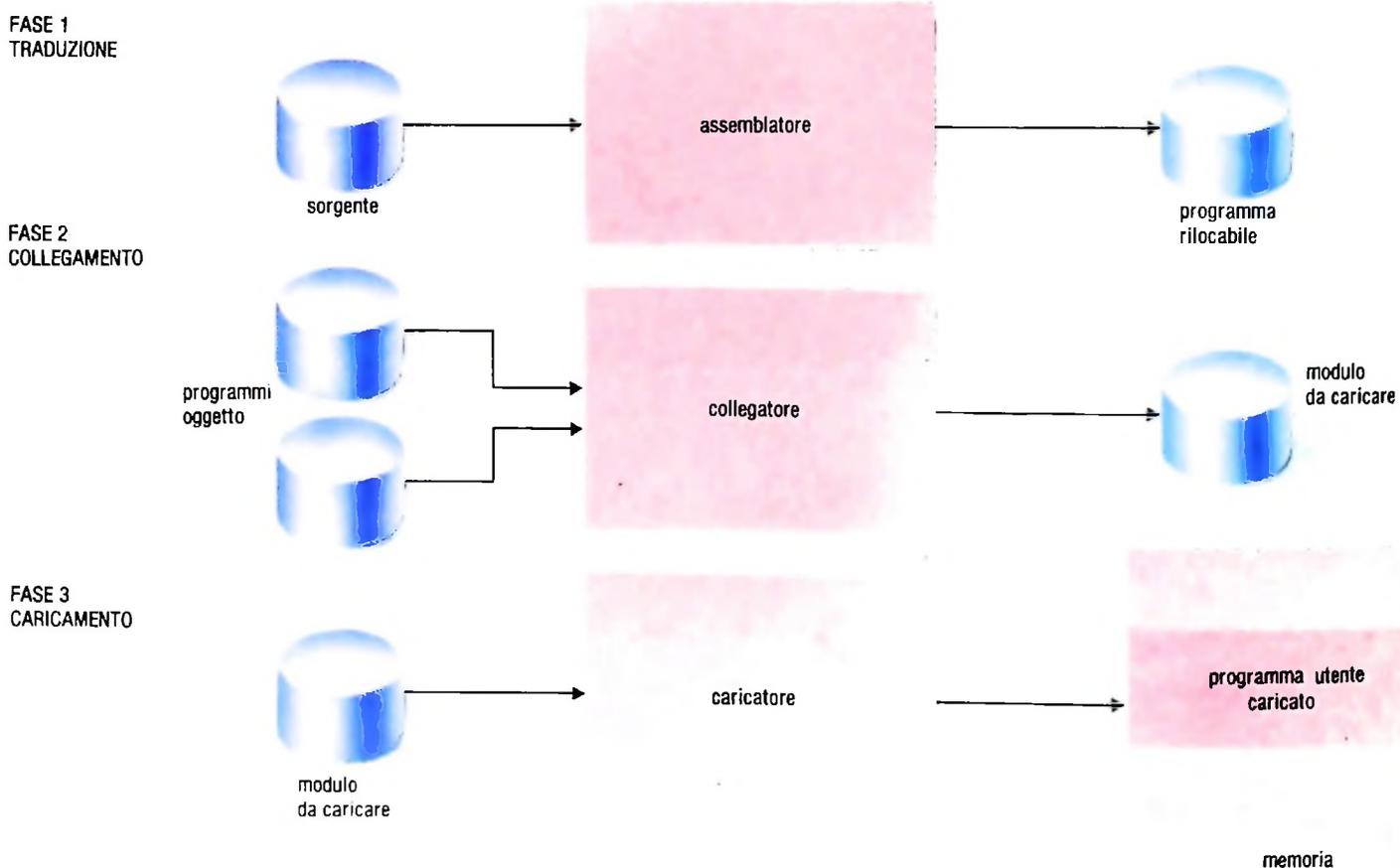
ma oggetto. Il compito quindi del collegatore è quello di analizzare tutte le informazioni sui simboli importati ed esportati e risolvere i riferimenti esterni simbolici rimpiazzandoli con l'indirizzo dell'oggetto.

Vediamo brevemente come lavora il programma collegatore. Esso prende in ingresso un certo numero di programmi oggetto rilocabili e analizza le informazioni simboliche presenti. Dipendendo dall'ordine in cui i moduli sono dati in ingresso, l'uso di un nome può precedere la sua definizione. Per questo motivo il collegatore lavora in due passi successivi, ognuno richiedente una scansione completa degli oggetti. Nel primo passo il collegatore scandisce ogni programma oggetto costruendo una tabella di tutti i posti dove i nomi esterni sono usati e una tabella delle locazioni associate a

ogni nome esportato. Quando tutte queste locazioni sono conosciute, il collegatore sostituisce, nel secondo passo, un indirizzo a ogni riferimento simbolico e costruisce un unico programma rilocabile, chiamato modulo da caricare. Questo è ottenuto appendendo sequenzialmente ogni modulo trattato e supponendo un indirizzo di caricamento di 0 (si veda, qui sotto, la prima illustrazione). Tutti i riferimenti simbolici sono quindi sostituiti da indirizzi rilocabili relativi all'inizio del modulo da caricare. Il risultato è un programma in forma rilocabile privo di riferimenti esterni simbolici e pronto per essere rilocato in memoria da un caricatore rilocante. L'intero processo di trasformazione di un programma da linguaggio sorgente simbolico a codice macchina direttamente eseguibile è sintetizzato nella figura più in basso.



A fianco: formato del modulo da caricare, costruito dal collegatore come unico programma rilocabile. Sotto: schema dell'intero processo di trasformazione del programma nel linguaggio-base accettabile dalla macchina.



DISCHI E NASTRI PER MEMORIZZARE

I principali supporti di memorizzazione e alcune norme per il loro impiego.

Il floppy disk è attualmente il supporto di memorizzazione più diffuso fra i personal computer, e vale la pena di spendere ancora due parole per esaminarne le caratteristiche e le modalità d'uso.

Il più comune criterio di valutazione dei floppy disk è la capacità di memoria espressa in Kbyte, che permette di avere istantaneamente una misura della quantità di informazioni disponibili in permanenza su un supporto magnetico. Conviene fare una distinzione netta fra floppy utilizzati per gli home computer e i portatili, e quelli usati con i personal, i micro e i minisistemi. I primi, per ragioni di costi globali, hanno prestazioni decisamente inferiori, ma più che sufficienti per la stragrande maggioranza delle applicazioni di ti-

po non professionale. Un altro parametro attendibile è la possibilità di collegamento al computer di più drive: avere a disposizione due o tre unità a disco significa in pratica raddoppiare o triplicare la capacità di memoria "in linea". La tabella comparativa mette a confronto questi dati riferiti a 46 computer appartenenti a tutte le categorie, con distinzione, dove occorre, fra floppy disk e hard disk.

Con i floppy, la parola d'ordine è: delicatezza. È necessario prestare una particolare attenzione nel maneggiarli, e per garantire la sicurezza dei dati è opportuno seguire alcune regole fondamentali che contribuiscono a preservare i dischetti. Ricordiamoci comunque che i floppy disk hanno un ciclo di vita relativamente breve, che si aggira intorno ai 10-12 mesi

TABELLA COMPARATIVA DRIVE

Home e portatili

| COMPUTER: | TIPO: | CAPACITA': | N. MASSIMO COLLEGABILE |
|-----------------|------------|------------|------------------------|
| Atari 600/800 | 5" | 127 KB | 4 |
| Casio FP 200 | 5" | 70 KB | 1 |
| CBM 64, VIC 20 | 5" | 170 KB | 4 |
| CBM 16, Plus 4 | 5" | 170 KB | 1 |
| Dragon 32 e 64 | 5" | 180 KB | 1 |
| Epson HX 20 | 5" | 320 KB | 4 |
| New Brain | 5" | 800 KB | 2 |
| HP 110 | 5" | 710 KB | 2 |
| Sharp M2 700 | 5" | 280 KB | 2 |
| Sharp PC 5000 | 5" | 320 KB | 2 |
| ZX Spectrum | microdrive | 100 KB | 8 |
| Sinclair QL | microdrive | 100 KB | 8 |
| TRS 80 CC 2 | 5" | 150 KB | 4 |
| TRS 80 mod. 100 | 5" | 184 KB | 2 |

La tabella comparativa che è pubblicata in questa pagina e nelle due successive mette a confronto 46 computer, indicando per ognuno di essi il numero di drive con cui è possibile eseguire un collegamento

Personal e micro

COMPUTER: TIPO: CAPACITA': N. MASSIMO COLLEGABILE

| | | | |
|---------------------|------|------------|---|
| Apricot PC | 3.5" | 315-720 KB | 2 |
| | Hard | 5-10 MB | 1 |
| Alpha Micro | 5" | 800 KB | 1 |
| | 8" | 1 MB | 1 |
| | Hard | 10-70 MB | 1 |
| Altos ACS 8000 | 8" | 500 KB | 1 |
| | Hard | 10-42 MB | 2 |
| Apple IIc | 5" | 140 KB | 2 |
| | Hard | 5 MB | 1 |
| Apple III | 5" | 140 KB | 3 |
| | Hard | 5-10 MB | 2 |
| Apple Macintosh | 3.5" | 400 KB | 2 |
| Apple Lisa II | 3.5" | 400 | 1 |
| | Hard | 5-10 MB | 7 |
| Canon AS 100 | 5" | 640 KB | 2 |
| | 8" | 1.2 MB | 2 |
| | Hard | 8.5 MB | 1 |
| CBM 600 e 700 | 5" | 170-534 KB | 2 |
| | Hard | 5-7.5 MB | 1 |
| Digital Rainbow 100 | 5" | 400 KB | 4 |
| | Hard | 10 MB | 1 |
| Epson FX 8 | 3.5" | 360 KB | 1 |
| | 5" | 360 KB | 1 |
| HP 150 | 3.5" | 270 KB | 2 |
| | Hard | 5-15 MB | 1 |
| IBM PC e XT | 5" | 180-360 KB | 2 |
| | Hard | 10 MB | 2 |
| Olivetti M20 | 5" | 320-640 KB | 2 |
| | Hard | 10 MB | 1 |
| Olivetti M21 | 5" | 360-720 KB | 2 |
| Olivetti M24 | 5" | 360-720 KB | 2 |
| | Hard | 10 MB | 1 |
| Philips P 20000 | 5" | 160-640 KB | 2 |
| | Hard | 10 MB | 1 |
| Sharp MZ 3541 | 5" | 386 KB | 2 |
| | 8" | 1 MB | 4 |
| | Hard | 10 MB | 1 |
| TI Professional | 5" | 320-640 KB | 2 |
| | Hard | 10 MB | 1 |
| Xerox X2000 | 5" | 800 KB | 2 |
| | Hard | 5-120 MB | 1 |

Mini

| | | | |
|---------------------|------|-----------|---|
| Alpha Micro AM10 XX | 5" | 800 KB | 1 |
| | 8" | 1 MB | 1 |
| | Hard | 70-400 MB | 1 |
| Burroughs B930 | 5" | 0.25-1 MB | 6 |
| | Hard | 18-77 MB | 3 |

| | | | |
|-----------------------|------|-------------|----|
| Digital PDP 11/24 | 5" | 400 KB | 2 |
| | 8" | 256-512 KB | 2 |
| | Hard | 5-28 MB | 4 |
| HP serie 3000 | Hard | 28-132 MB | 28 |
| Honeywell DPS 6/75 | 5" | 650 KB | 8 |
| | 8" | 512 KB | 8 |
| | Hard | 20+20 MB | 4 |
| IBM serie I | 5" | 300-1200 KB | 2 |
| | Hard | 9-200 MB | 1 |
| Olivetti M30-40 | 5" | 1 MB | 2 |
| | 8" | 1 MB | 4 |
| | Hard | 18 MB | 4 |
| Olivetti M60 | Hard | 60-120 MB | 1 |

per un uso quotidiano. Ecco un decalogo delle operazioni da compiere e da evitare usando i floppy disk.

1 - Riporre sempre i dischi nella custodia dopo l'uso. La polvere è il nemico peggiore della delicata superficie magnetica che riveste il dischetto. La guaina interna del contenitore è ricoperta da uno speciale materiale autopulente, che trattiene la polvere ed evita graffiature accidentali del disco durante la rotazione. Tuttavia, è bene non esporre i preziosi dati memorizzati sul floppy a rischi eccessivi.

2 - Non toccare mai la superficie del disco visibile attraverso la fessura di accesso. L'apertura permette alla testina di lavorare sul disco in rotazione. Le dita lo sporcheranno, con grave rischio per l'integrità dei dati.

3 - Non curvare o piegare i floppy. Il cartoncino con cui è costruita la custodia è abbastanza rigido, ma non assicura una protezione totale. In questo senso, i microfloppe da 3 pollici e mezzo sono molto più sicuri, perché il contenitore in plastica ne garantisce l'indistruttibilità.

4 - Non cercare di pulire i floppy, perché si rischia di danneggiare la superficie magnetica.

5 - Non esporre i dischi a fonti di calore o a forti campi magnetici. In particolare, è bene evitare di appoggiare i floppy sul monitor o sul televisore, per non rischiare di ritrovarli... formattati automaticamente.

6 - Non scrivere sul floppy con matite appuntite o penne biro che potrebbero danneggiare il disco interno. Usare invece pennarelli di feltro a punta morbida.

7 - Non accendere o spegnere il computer o il drive con il disco inserito: le correnti transienti che intervengono in apertura e chiusura del circuito di alimentazione potrebbero creare un forte campo magnetico che smagnetizza il disco.

8 - Riporre al sicuro le buste vuote dei floppy durante l'uso, per evitare la polvere.

9 - Non maneggiare liquidi in prossimità del computer o dei floppy, ed evitare la vicinanza di sigarette e posacenere. Anche le microparticelle di fumo possono danneggiare i dati memorizzati sul floppy.

10 - Fare regolarmente copie dei dischi più importanti e di uso frequente. Utilizzare le copie a rotazione, per rendere omogenea l'usura e diminuire il rischio di perdita dei dati.

Hard disk

I floppy disk, soprattutto quelli da cinque pollici e un quarto, rappresentano un ottimo compromesso tra costo e prestazioni nell'ambito dei personal e degli home computer.

Esistono però numerosi casi in cui la quantità di dati che deve essere disponibile "in linea" in un dato momento supera tranquillamente il Megabyte, ponendosi quindi ben al di sopra dell'attuale limite della tecnologia dei floppy disk.

In tutte quelle situazioni in cui sia necessario accedere a grosse quantità di dati in un tempo molto breve e senza dover continuamente cambiare i dischetti nel drive, si impiega quindi un altro tipo di memoria di massa: il disco rigido o "hard disk".

A prova di polvere

La quantità di dati memorizzabile su di un floppy dipende, come abbiamo già visto, dalla densità di informazioni che la testina del drive, sotto la guida del sistema operativo, è in grado di "scrivere" per ogni unità di superficie magnetica del disco stesso. La superficie di disco occupata fisicamente da un bit su un disco a "doppia densità" sarà quindi la metà rispetto a quella occupata dallo stesso bit su di un disco a "singola densità".

Facendo uso di meccanismi molto sofisticati e di tolleranze bassissime, sarebbe perciò possibile compattare su un singolo disco quantità molto elevate di informazioni, scrivendo ogni bit su una sezione piccolissima del disco.

Questa possibilità teorica si scontra però con una limitazione di carattere pratico. Più piccola è infatti la superficie magnetica occupata da un bit e più piccolo sarà il campo magnetico da essa generato e, di conseguenza, più vicina alla superficie del disco stesso dovrà essere la testina di lettura per consentire una acquisizione di dati priva di errori.

La distanza disco-testina non può però essere ridotta eccessivamente in quanto un granello di polvere, presente in qualsiasi ambiente, potrebbe incastrarsi tra le superfici in movimento causando danni irreparabili.

La soluzione a questo problema è stata trovata agli inizi degli anni Settanta dall'IBM, che realizzò un nuovo tipo di memoria di massa in cui sia la testina sia il disco (in materiale rigido, per assicurare la necessaria stabilità) erano racchiusi in un contenitore a chiusura ermetica nel quale l'aria, necessaria per compensare eventuali variazioni dimensionali dovute all'escursione termica, veniva ammessa solamente dopo una accuratissima filtrazione.

Così si poté avvicinare la testina al disco fino a distanze relative di pochi millesimi di millimetro, ottenendo densità di informazione altissime.

Le memorie di massa che sfruttano questa tecnologia sono chiamate anche "dischi rigidi" o "dischi fissi", indicando in questo modo che il supporto della memoria non è, a differenza del floppy, intercambiabile, oppure "Winchester", dal nome in codice che era stato assegnato al progetto "hard disk" da parte dei responsabili della IBM.

Piccoli, capaci e velocissimi

Similmente a quanto è avvenuto per i floppy, anche le dimensioni degli hard disk sono andate calando, mentre la loro capacità di memorizzazione è cresciuta notevolmente.

Il diametro dei dischi all'interno dei Winchester è passato in pochi anni da otto pollici a cinque, per arrivare, in alcuni casi, a tre pollici e nove decimi. La capacità di memorizzazione va da circa cinque Megabyte, per i modelli collegabili ai vari personal computer, a oltre 350 Megabyte per gli hard disk che lavorano con mini e microsistemi.

Oltre ad offrire una grande quantità di spazio, gli hard disk consentono anche un accesso rapidissimo alle informazioni in essi contenute e una velocità di trasferimento dati notevolmente più elevata rispetto ai migliori floppy.

Il tempo medio di accesso a un qualsiasi dato si aggira infatti per queste apparecchiature intorno ai 150-200 millisecondi, mentre i dati possono essere inviati alla RAM a velocità dell'ordine dei Megabyte al secondo.

Hard + nastro = sicurezza

Nell'hard disk tutte le informazioni e i programmi risiedono fisicamente sullo stesso disco. L'eventualità del danneggiamento di un Winchester non può essere esclusa e, se si dovesse verificare, potrebbe causare all'utente danni irreparabili. Non essendo realizzabile, salvo in alcuni casi particolari, un backup su floppy disk, l'uso di un Winchester implica l'impiego di un altro tipo di sistema di sicurezza chiamato "data streamer".

Il data streamer non è altro che un registratore molto sofisticato che, facendo uso di apposite cartucce magnetiche, è in grado di assicurare un'alta densità di dati per pollice di nastro, tale da consentire la copia anche di più hard disk sulla medesima cartuccia. I data streamer più diffusi fanno uso di nastri da un quarto di pollice (da 20 a 90 Megabyte per cartuccia) o da mezzo pollice (oltre 100 Megabyte).

Nastri magnetici

Abbandonate le schede perforate e i nastri di carta, nei moderni centri di calcolo è difficile trovare sistemi di lettura/scrittura dati simili a quelli utilizzati dai personal e dagli home computer. Tra le unità di memorizzazione più utilizzate si trovano i nastri magnetici sui quali vengono copiati in formato sequenziale gli archivi più importanti.

Il nastro è costituito da un supporto di plastica largo poco più di un centimetro (12,5 mm per l'esattezza) e di lunghezza variabile. Il nastro è rivestito, nella parte rivolta verso la testina di lettura/scrittura, da un sottile strato di materiale ferromagnetico analogo a quello che ricopre i floppy disk e i normali nastri per musicassette. I nastri, avvolti su bobine con attacco standard, vengono letti e scritti da apposite unità periferiche del computer centrale.

Le unità a nastro sono dotate di due testine (una per la lettura e una per la scrittura) e di un complesso meccanismo di trascinamento e riavvolgimento del nastro magnetico, che può raggiungere tranquillamente i tre chilometri di lunghezza. Il nastro è suddiviso in più piste parallele (dette anche tracce) che vengono magnetizzate dalla testina di registrazione. Le piste, che possono essere anche nove, accolgono i dati in forma binaria provenienti dall'elaboratore. La quantità di informazione memorizzabile sul nastro varia in funzione della lunghezza del nastro e della densità di registrazione dei dati per unità di superficie. Un nastro standard da mezzo pollice di lunghezza media è in grado di memorizzare alcune decine di milioni di caratteri.

Disk pack

Un tipo di memoria di massa che consente notevoli densità di informazione per unità di superficie e altissime velocità di accesso è rappresentato dal "disk pack", o pila di dischi, usato principalmente sui grossi elaboratori. Un disk pack consiste essenzialmente in un numero variabile di dischi di eguale diametro che girano impilati sul medesimo asse.

Le testine di lettura/scrittura, due per ogni disco (faccia superiore e faccia inferiore), si muovono solidalmente le une alle altre, in quanto collegate al medesimo braccio.

In un dato istante, quindi, le varie testine "punteranno" la medesima traccia di ciascun disco.

Tecnicamente l'insieme delle tracce che si trovano nella medesima posizione su tutti i dischi si chiama "cilindro". I dati presenti su un cilindro possono quindi essere letti e scritti senza dover muovere le testine, consentendo in tal modo notevoli risparmi di tempo.

Un altro tipo di disk pack fa invece uso di una testina fissa per ogni traccia di ciascun disco. In questo modo si ottiene, sia pure a prezzo di una maggiore complicazione costruttiva, una ulteriore riduzione dei tempi di accesso, dovuta alla eliminazione degli spostamenti delle varie testine.

La capacità di un modulo disk pack può arrivare ai 500-600 Megabyte, mentre la velocità di trasferimento dati è nell'ordine dei due Megabyte al secondo.

Lezione 48

Curve sovrapposte

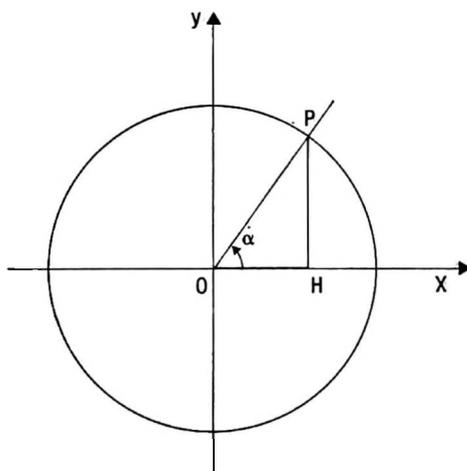
Completiamo i grafici fatti nell'ultima lezione aggiungendo alla funzione seno quella "coseno".

Così come abbiamo fatto per il seno, vediamo la definizione di questa nuova funzione, che già è stata usata per tracciare circonferenze, e che ancora useremo a tale scopo con il nostro microplotter.

Riprendiamo la nostra circonferenza con il centro sull'origine di due assi perpendicolari x e y (chiameremo, seguendo la consuetudine dei matematici, l'asse x come "asse delle ASCISSE" e l'asse y come asse delle "ORDINATE").

Consideriamo una semiretta uscente dall'origine, che forma un angolo α con l'asse delle ascisse; essa interseca la circonferenza in un punto P , che ha come ascissa il segmento OH .

Si definisce "coseno" dell'angolo il rapporto tra il segmento OH e il valore del raggio della circonferenza.



Così come il seno, anche il coseno assume valori che variano tra -1 e 1 : vale -1 per un angolo di 180 gradi (infatti in tale situazione il segmento OH diventa una misura negativa in termini di ascissa) e vale 1 per un angolo nullo, caso in cui P si trova sull'asse delle ascisse, nella parte positiva dello stesso.

Anche per questa funzione vale la considerazione sull'unità di misura normalmente usata dai matematici per gli angoli: invece di misurarli in gradi, useremo il radiante, ricordando che un angolo giro di 360 gradi ha il valore di 2π radianti.

Anche in questo caso, la funzione coseno è messa a disposizione dal BASIC, e noi dovremo fornire l'argomento espresso in radianti.

Per vedere con chiarezza come varia tale funzione, possiamo, ancora una volta, tracciarne il grafico con il nostro microplotter.

Anzi, tracciandolo insieme al grafico precedentemente prodotto che indica l'andamento del seno, potremo confrontarne le caratteristiche.

Il programma precedentemente costruito non cambia molto: si tratta in sostanza di aggiungere solo una parte che traccia questa nuova funzione, che potrà seguire completamente sul modello già adottato per il seno. Tuttavia, per una maggiore "pulizia" di immagine e di concetti, apporteremo le seguenti modifiche:

- poiché la funzione seno era tracciata in rosso, ne scriveremo il nome sul foglio in rosso;
- tratteremo la funzione coseno in verde, e scriveremo il relativo nome sotto a quello del seno, sempre in verde;
- introdurremo alcuni comandi "H", per eliminare il tracciamento non desiderato di alcune linee che "sporcano" l'immagine prodotta dal programma visto nell'ultima lezione. Al lettore attento non sarà infatti sfuggito che, dopo aver tracciato gli assi, il fatto di cambiare colore e di chiedere il tracciamento della funzione seno comportava la sovrapposizione di una linea rossa dal punto in cui era stato terminato il tracciamento dell'asse fino all'origine, ove iniziava il tracciamento del seno. È necessario quindi introdurre un comando di tale tipo prima del tracciamento del seno, ed analogamente, prima del coseno.

La lista del programma risulta quindi la seguente:

```

10 LPRINT CHR$(18)
15 LPRINT "X1,120,4" 'Asse x
16 LPRINT "M0,-150 'Asse y
17 LPRINT "X0,50,6"
18 LPRINT "H" 'Per eliminare traccia
19 LPRINT "C3" 'Rosso
20 FOR I=0 TO 6.28 STEP .1
30 LET P$="D"+STR$(I*480/6.28)+", "+STR$(
SIN(I)*100)
40 LPRINT P$
50 NEXT I
60 REM Scala
65 LPRINT "C1" 'Blu
70 LPRINT "M4,-18" 'Origine
80 LPRINT "P0"
90 LPRINT "M106,-18" 'Pi mezz i
100 LPRINT "PPI/2"
110 LPRINT "M230,-18" 'Pi
120 LPRINT "PPI"
130 LPRINT "M340,-18" '3/2 Pi
140 LPRINT "P3/2 Pi"
150 LPRINT "M438,5" '2 Pi
155 LPRINT "P2 Pi"

```

```

160 LPRINT "M5,95"      'Asse y
170 LPRINT "P+1"
180 LPRINT "M5,-105"
190 LPRINT "P-1"
200 LPRINT "M3,150"    'Coordinata y
210 LPRINT "Py"
220 LPRINT "M470,-20"  'Coordinata x
230 LPRINT "Px"
240 LPRINT "M300,120"  'Nome funzione
250 LPRINT "S2"        ' Cambia misura
260 LPRINT "C3"        ' In rosso
270 LPRINT "Py=sin(x)"
275 LPRINT "C2"        'Verde per il coseno
277 LPRINT "H"        'Elimina traccia
280 FOR I=0 TO 6.28 STEP .1
290 LET P$="D"+STR$(I*480/6.28)+", "+STR$(
(COS(I)*100)
300 LPRINT P$
310 NEXT I
320 LPRINT "M300,100"  'Nome funzione
330 LPRINT "Py=cos(x)"

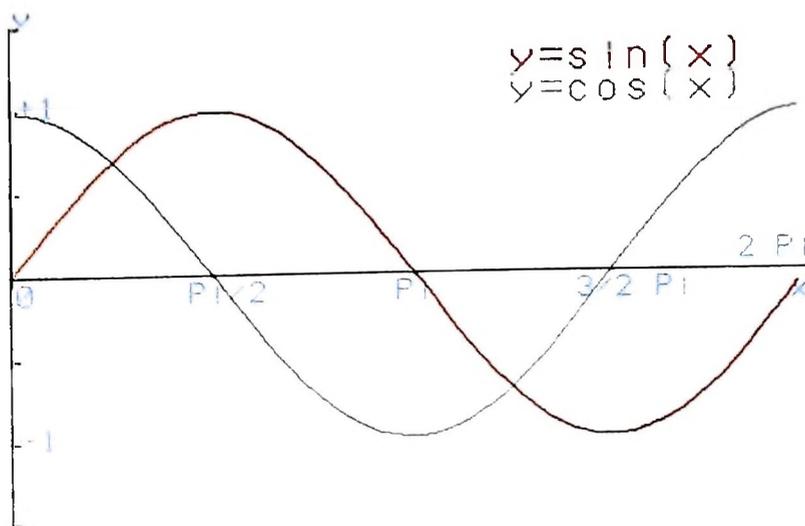
```

ove le istruzioni alle linee 18 e 277 hanno lo scopo di evitare che venga lasciata una traccia quando viene iniziata una nuova parte del grafico.

Le linee dalla 275 in poi sono equivalenti a quelle per tracciare il seno, usate per la funzione coseno.

Il tracciamento degli assi e delle varie scritte a fianco degli stessi ha mantenuto, ovviamente, validità.

Eseguiamo ora il programma, e otteniamo:



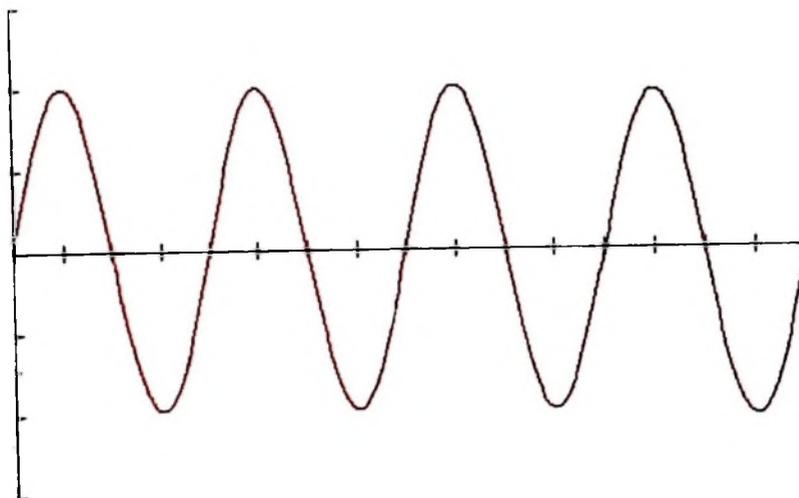
Si può osservare che abbiamo ottenuto lo scopo che ci eravamo prefissi; in particolare, possiamo osservare che le due curve sono identiche, a parte il fatto che sono sfasate tra loro di un valore di $\pi/2$: ovvero, il seno di un certo angolo x è uguale al coseno di un angolo $x - \pi/2$.

Le funzioni che abbiamo visualizzato sono PERIODICHE: dopo un tratto di ascissa $2*\pi$, il loro comportamento si ripete.

Possiamo evidenziare questo fatto con il programma:

```
10 LPRINT CHR$(18)
12 LPRINT "X1;30,16" 'Asse x
15 LPRINT "M0,-150"
20 LPRINT "X0,50,6" 'Asse y
25 LPRINT "H"
30 LPRINT "C3" 'Rosso
45 FOR I=0 TO 4*6.28 STEP 0.1
50 LET P$="D"+STR$(I*120/6.28)+"", "+STR$(
SIN(I)*100)
55 LPRINT P$
60 NEXT I
```

che causa la visualizzazione seguente:



Cosa abbiamo imparato

In questa lezione abbiamo visto:

- il significato della funzione COSENO;
- le caratteristiche di questa funzione rispetto alla funzione seno.

LE WORKSTATION

Che cosa sono e come funzionano le "stazioni di lavoro".

Nella seconda metà degli anni Settanta si è sviluppato ed affermato un nuovo modo di concepire il calcolatore personale. Questo grazie particolarmente al lavoro di Alan Kay che all'interno del Learning Research Group presso il Palo Alto Research Center della Xerox ha contribuito all'affermazione di sistemi di calcolo di nuova concezione. Si sviluppò quindi l'idea di quello che venne allora definito un Personal Dynamic Media, che può senz'altro essere considerato uno dei primi passi verso l'implementazione dei sistemi definibili come workstation.

Con il termine workstation si suole indicare una stazione di lavoro monoutente, dotata di potenza di memoria e di calcolo nettamente superiori a quelli usuali per un personal computer. La stazione di lavoro è corredata da strumenti software integrati in un contesto unico, da cui in modo estremamente flessibile si può accedere a tutte le risorse disponibili. L'integrazione delle risorse software costituisce l'elemento chiave perché l'interazione con il computer sia di tipo completamente nuovo.

In un ambiente tradizionale (non integrato) l'utente dispone

di una serie di funzionalità garantite da altrettanti programmi applicativi.

Il sistema operativo presiede, tra l'altro, alla gestione di questi e in una visione tradizionale tende a mantenere rigidamente separati gli ambienti di ogni applicazione.

Spesso inoltre all'interno di una stessa applicazione esistono diversi modi di agire cui corrispondono altrettanti ambienti indipendenti, per esempio i modi *insert*, *delete*, *copy* presenti all'interno di un Editor.

Un esempio servirà a chiarire questi concetti: l'utente crea un programma la cui esecuzione produrrà la visualizzazione di un disegno sullo schermo del terminale. All'inizio, dal livello più alto, quello di sistema, l'utente può scegliere quale funzionalità attivare, e in questo caso entrerà nell'ambiente Editor per la scrittura del codice del proprio programma; terminata la stesura del programma, bisognerà uscire dall'ambiente Editor e attivare il Compilatore a cui viene passato il file contenente il codice appena scritto. In assenza di errori l'utente "entrerà" in modalità grafica per eseguire finalmente il programma compilato. In tutte queste fasi all'utente



Ormai il calcolatore si va adattando a sempre nuove e specifiche esigenze di lavoro.

vengono presentati un tipo di informazione e di ambiente in cui non è possibile verificare quali interferenze e modifiche le azioni lì svolte comportano in altri ambienti.

Nel caso in cui il risultato ottenuto non sia quello desiderato, la catena di azioni prima descritte dovrà essere ripercorsa per permettere le operazioni di correzione. Viene così persa la visualizzazione del risultato finale, su cui l'utente dovrà agire tenendo a mente le modifiche necessarie e agendo in maniera indiretta in un ambiente diverso da quello in cui è collocato l'oggetto da modificare.

Nella situazione peggiore nei vari ambienti associati alle diverse funzionalità del sistema, i comandi di livello più alto non saranno direttamente disponibili, così che, per esempio, la stampa dei risultati potrebbe richiedere il ritorno al livello sistema per invocare la funzione di Print.

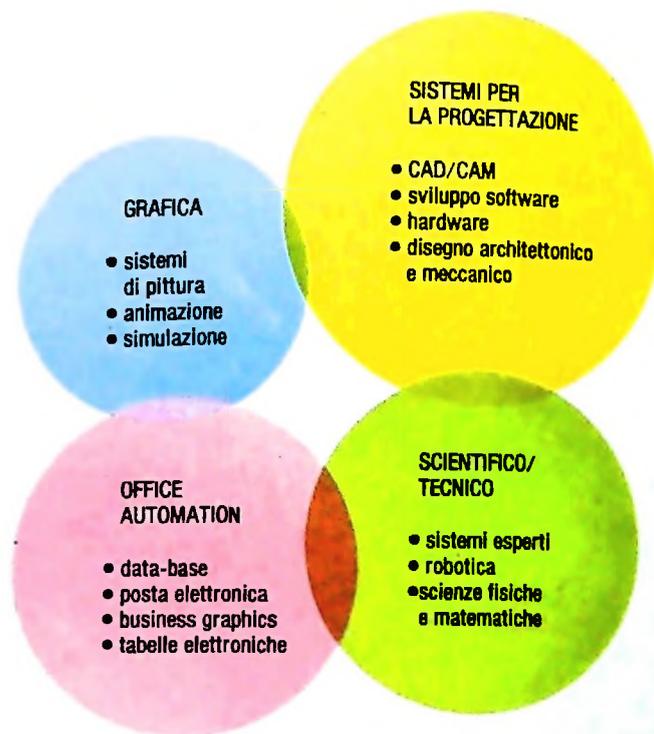
La realizzazione di un ambiente software integrato costituisce la soluzione di questo genere di problemi permettendo, in un certo senso, di "avere tutto sempre sottomano" ed eliminando quindi percorsi tortuosi tra le funzionalità del sistema. La metafora più utilizzata e che forse rende meglio il senso di questi discorsi è quella della scrivania, su cui sono sparsi e sovrapposti molti fogli di carta, ognuno dei quali serve a supportare parti di un'unica attività. Sulle moderne workstation i fogli di carta corrispondono alle cosiddette finestre, il cui comportamento è assolutamente analogo.

Per la realizzazione di questo obiettivo è essenziale, quindi, la presenza di un sistema per la gestione sullo schermo di più finestre (window system). Ciò equivale a poter disporre di terminali virtuali, in ognuno dei quali vengono eseguite attività differenti. Le finestre occuperanno porzioni di schermo e potranno variamente sovrapporsi. In ogni momento sarà possibile portare in cima alla pila la finestra di interesse utilizzando menù e dispositivi di puntamento sullo schermo, come il mouse, che sostituiscono validamente l'uso dei tasti funzionali.

Dal punto di vista hardware questo implica che lo schermo sia di tipo raster (a matrici di punti) e che ad esso corrisponda, quindi, quella che viene detta una bit-map in memoria. In tal modo a ogni punto dello schermo (pixel) è associato un bit o una configurazione di bit. Manipolando opportunamente porzioni della bit-map è possibile gestire facilmente delle porzioni dello schermo attraverso operazioni di tipo booleano, su matrici di bit, che rappresentano quindi le corrispondenti operazioni sul raster, ovvero operazioni sulle finestre. Queste operazioni consisteranno principalmente in spostamenti, modificazioni delle dimensioni e sovrapposizioni, che devono essere tutte gestite in modo tale da non perdere o alterare le informazioni che sono contenute in ciascuna finestra.

Tenuto conto che uno schermo a media risoluzione equivale ad una matrice di 512x512 pixel, mentre in uno ad alta risoluzione si arriva fino a 1024x1024 e che ad ogni pixel sono associati da 1 fino a 24 bit, si capisce come una bit-map necessiti di grandi quantità di memoria e la quantità di calcoli da eseguire per gestire ogni singolo pixel sia notevole.

Un altro aspetto rilevante consiste nella disponibilità per il singolo utente di potenze di calcolo riservate prima ad una



classe di sistemi come quella dei minicomputer, senza per altro dover sopportare i possibili ritardi nei tempi di risposta, tipici di un sistema a divisione di tempo.

In un sistema a divisione di tempo molti terminali sono collegati ad una macchina dotata di una unica CPU che distribuisce la propria attività tra i processi associati a ciascuno dei terminali connessi. L'attività della CPU è completamente trasparente per il terminale e per l'utente che lo utilizza, che quindi lavora come se avesse a disposizione una macchina personale.

Questo è tanto più vero quanto più la CPU è potente, quindi in grado di sopportare più terminali, e quanto minore è il numero di terminali collegati; in una workstation invece il rapporto CPU/utente è 1 a 1.

Le risorse che possono comunque essere condivise da molte workstation saranno dischi, stampanti, plotter, unità a nastro ecc., realizzando in tal modo interessanti e non trascurabili economie di scala.

La possibilità di avere a disposizione hardware più potenti a

costi minori di quelli necessari, nel recente passato, per disporre delle stesse potenze di calcolo, ha costituito inoltre la base per la realizzazione di migliori interfacce uomo/macchina. Con questo si intende una combinazione di hardware e software, capace di integrare semplicemente funzionalità grafiche, di trattamento dei testi, e di computazione vera e propria, caratteristiche, queste, presenti in misura diversa su tutte le workstation attualmente sul mercato.

Nell'illustrazione di queste pagine si è cercato di evidenziare la varietà dei settori coinvolti come possibili fasce di utenza di sistemi workstation, e che sono soprattutto quelli scientifico-tecnici, di progettazione e di grafica avanzata.

Il problema degli standard

Nonostante che per i sistemi esistenti sul mercato la generale tendenza delle case costruttrici sia quella di perseguire indirizzi propri, esistono per un sistema workstation alcuni ele-

La workstation (vera e propria "stazione di lavoro") è corredata da strumenti software integrati in un contesto unico, da cui si può accedere a tutte le risorse disponibili. Questo coinvolge diversissime e sofisticate fasce di utenza.



menti di base comuni relativamente ai quali tutti cercano di tener conto degli standard industriali esistenti, sia hardware che software. Questa attenzione agli standard è un elemento che determina molto spesso il successo di un prodotto e la sua accettazione da parte della comunità informatica. Il rispetto di alcuni standard garantisce la massima compatibilità di un sistema con l'ambiente esterno, ovvero per esempio la possibilità di collegarsi a reti di comunicazione già esistenti, o di espandere l'hardware a disposizione.

Gli standard vengono in generale fissati o riconosciuti da organismi sia nazionali che internazionali, che spesso tendono a confermare quello che è già stato stabilito da una situazione di fatto di mercato.

I settori principali in cui questi organismi operano, alcuni dei quali verranno di seguito trattati, sono:

- le funzionalità grafiche;
- la comunicazione tra calcolatori;
- la comunicazione all'interno di un calcolatore;
- i sistemi operativi;
- i linguaggi;
- l'hardware per il calcolo numerico.

Le caratteristiche principali di una workstation

La tendenza generale è verso l'utilizzo di processori a 32 bit, quindi con un ampio spazio di indirizzamento e con potenza di calcolo sufficiente a supportare interfacce sofisticate con gestione di molte finestre sullo schermo, sistemi operativi multitasking, terminali video a bit-map, uso di dispositivi di puntamento come il mouse ecc.

La connessione dell'hardware a bus standard garantisce maggiormente la sua espandibilità ovvero la possibilità di fare uso delle miriadi di schede che il mercato offre per le più svariate applicazioni.

In questo campo si è passati da una situazione in cui le case costruttrici di hardware progettano sistemi completi, indipendentemente da qualsiasi convenzione, ad una situazione di specializzazione nella produzione esclusiva di schede su cui erano già predefiniti dei canali di comunicazione, o di schede di CPU o di memoria.

All'affermazione quindi di sistemi sostanzialmente aperti, ha fatto seguito la necessità di fissare degli standard a cui tutti potessero far riferimento per la connessione delle proprie schede a quelle di altri costruttori.

Uno degli standard più affermati per i bus di comunicazione è il IEEE-796, meglio noto come MULTIBUS, che permette il passaggio di indirizzi fino a 20 bit (corrispondente all'indirizzabilità diretta di 1 Megabyte di memoria) che possono essere espansi fino a 24. Nonostante ciò emerge già l'esigenza di capacità superiori verso i 32 bit di indirizzo.

L'uso di una workstation in settori tecnico-scientifici comporta spesso una quantità ingente di calcoli numerici, per i quali è prevista l'esistenza di appositi coprocessori aritmetici, dedicati per esempio alle operazioni in virgola mobile (floating point). Questi alleggeriscono il processore centrale da un carico che potrebbe risultare troppo pesante in alcune appli-

cazioni, e che genererebbe un rallentamento delle prestazioni e dell'efficienza dell'intero sistema. Tra gli standard che in questo campo sono emersi, in particolare per le operazioni floating point, vi è il IEEE-764 con cui ci si riferisce ad un numero con segno di 32 bit, di cui 8 bit di esponente e 23 bit di mantissa, oppure a un numero con segno di 64 bit di cui 11 di esponente e 52 di mantissa.

Uno degli elementi chiave per questo come per altri sistemi è il software. Per esempio l'esistenza di più finestre ad ognuna delle quali associare un'attività richiede un sistema operativo in grado di sopportare più processi contemporaneamente.

Per quanto riguarda il software è importante tenere conto del suo livello di portabilità e quindi della esistenza di sistemi operativi che supportino linguaggi di programmazione ad alto livello, il più possibile standard e diffusi.

I principali obiettivi per il software sono essenzialmente i seguenti:

- indipendenza dal particolare tipo di terminale, ovvero possibilità di utilizzare con un'applicazione diversi tipi di periferiche di output;
- portabilità dei programmi, ovvero indipendenza dei pacchetti applicativi da un hardware particolare.

In campo grafico, per esempio, si sono affermate alcune tendenze e i principali standard grafici presenti oggi sul mercato sono il CORE System prodotto dal Graphics Standards Planning Committee dell'americana ACM (Association for Computing Machinery), e il Graphical Kernel System (GKS) sviluppato invece in Europa. Al di là delle caratteristiche proprie dei due sistemi, entrambi permettono lo sviluppo di applicazioni grafiche il più possibile omogenee e trasportabili su hardware diversi.

La necessità di accedere ad informazioni remote, o l'esigenza di comunicazione locale, ha comportato lo sviluppo di reti geografiche alle quali si accede per collegarsi, per esempio, a banche dati, e lo sviluppo di reti locali che permettono la comunicazione tra più workstation collocate nello stesso ambiente di lavoro.

Per quanto riguarda le reti locali uno degli standard più diffusi è sicuramente ETHERNET, proposto dalla Xerox, per cui molte workstation forniscono opportune interfacce di collegamento.

Solo recentemente l'International Standards Organization è arrivata a definire la struttura generale che deve servire da modello per la realizzazione di una rete generica. Questo modello è organizzato in 7 livelli per ognuno dei quali è definita una particolare interfaccia con i livelli adiacenti e quindi precisi protocolli di comunicazione tra i livelli stessi.

Questi livelli servono a distinguere i diversi elementi che costituiscono un processo di comunicazione, a cui corrispondono altrettanti livelli di astrazione in cui strutturare questi particolari tipi di sistema: si va dal livello fisico, con riferimento ai segnali elettrici che viaggiano lungo i cavi di collegamento, fino al livello utente in cui vengono formulati i messaggi e ricevute le informazioni.

Rispettare questo standard significa che la workstation deve essere dotata dell'hardware e del software necessario a gestirlo opportunamente.

LE COORDINATE OMOGENEE

Come ottenere una trasformazione tridimensionale attraverso la tecnica del calcolo matriciale.

Nel precedente articolo abbiamo introdotto per sommi capi le problematiche relative alla terza dimensione e abbiamo delineato il concetto di "proiezione", articolandolo nelle due categorie fondamentali rappresentate dalle "proiezioni prospettiche" e dalle "proiezioni parallele". In questo numero vogliamo fornire un approccio un po' più rigoroso al problema, introducendo il concetto di coordinate omogenee e di trasformazioni tridimensionali mediante la tecnica del calcolo di matrici. Segnaliamo sin da subito che oltre alle trasformazioni geometriche già viste per il caso bidimensionale, anche le proiezioni si prestano a una interpretazione matriciale.

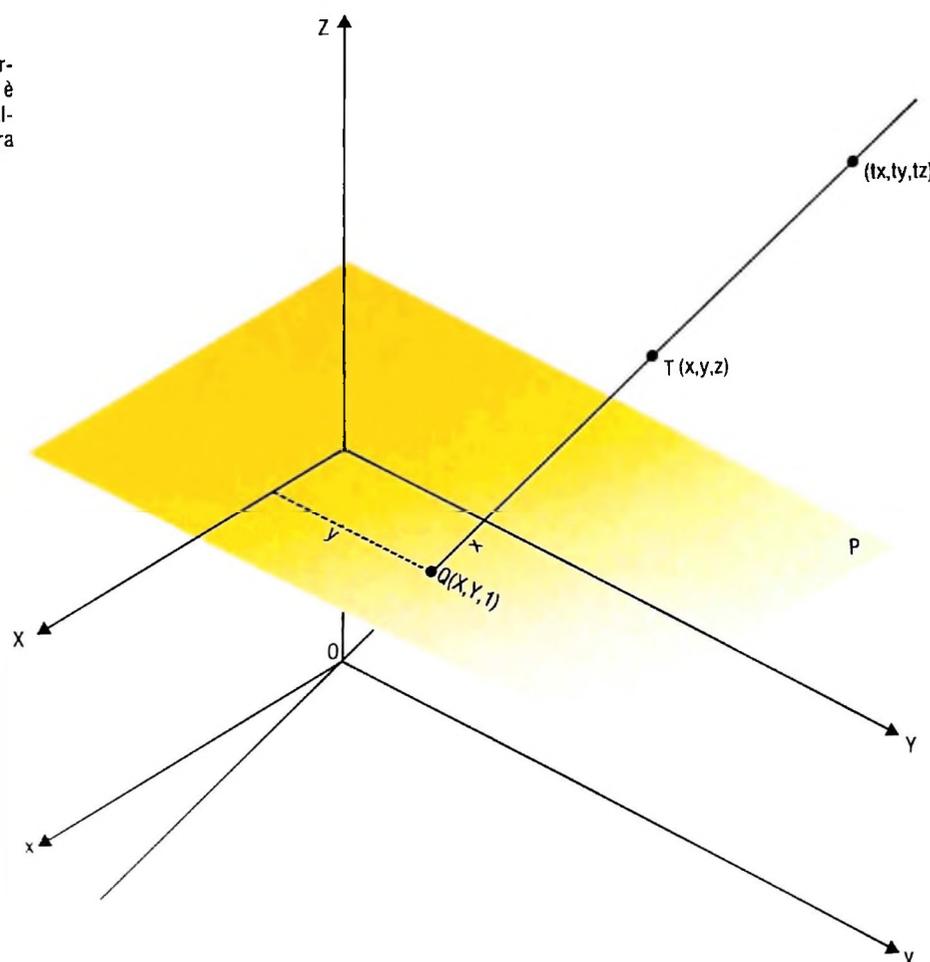
Le coordinate omogenee

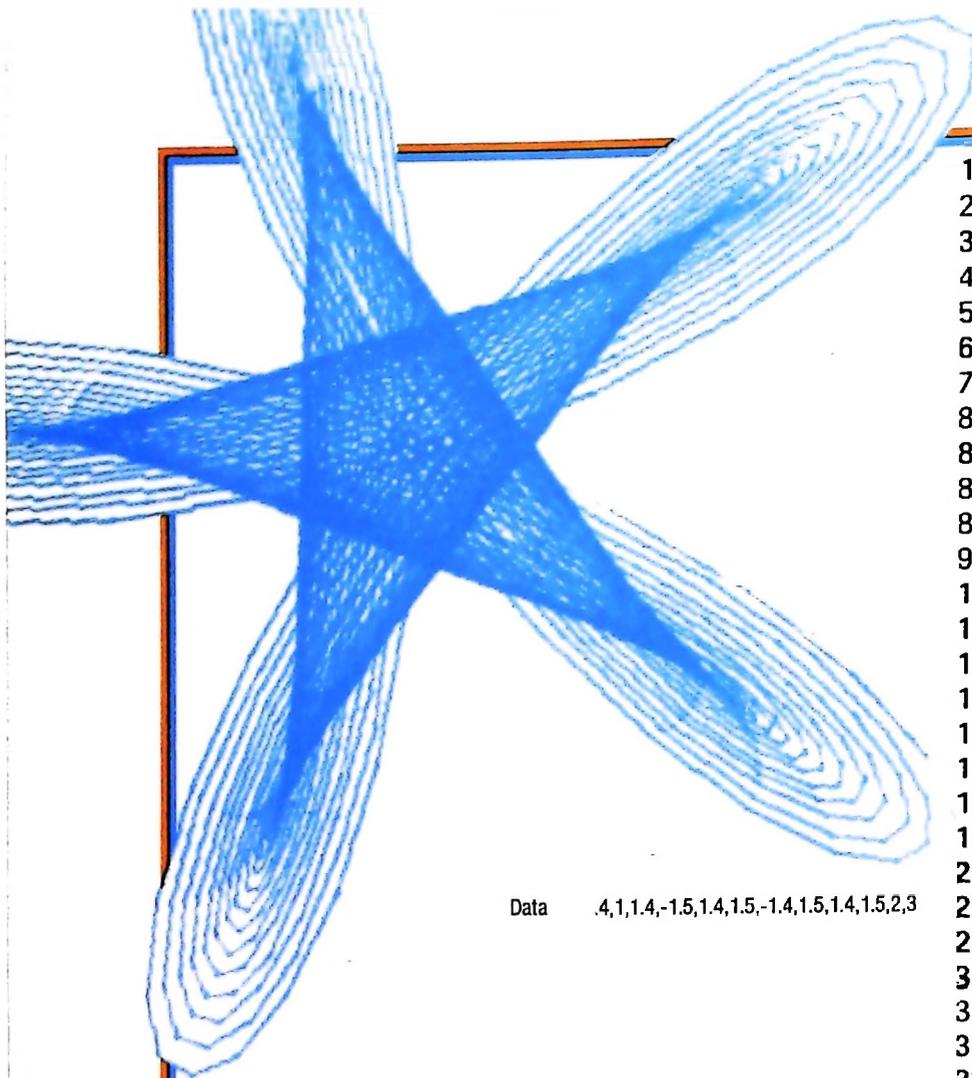
Due domande sorgono immediatamente spontanee: cosa sono e perché si usano le coordinate omogenee?

Per comodità e chiarezza espositiva inizieremo col dare una risposta alla seconda domanda, rifacendoci a quanto visto negli articoli sulle trasformazioni geometriche per il caso bidimensionale. In questo contesto abbiamo verificato che una matrice quadrata del tipo 3×3 , ossia costituita da 3 righe e da 3 colonne, permette di esprimere generiche trasformazioni di roto-traslazione e di scala in maniera unificata. L'artificio formale era allora consistito nell'aumentare di uno la dimensione dei vettori, ossia nel passare da un vettore $[x \ y]$ a due componenti, a un vettore $[x \ y \ 1]$ a tre componenti: questo viene detto "vettore omogeneo" e "coordinate omogenee" le tre componenti. L'utilità pratica di questa estensione del vettore bidimensionale sta nel fatto che in forma omogenea qualsiasi vettore è "conformabile", ossia è possibile eseguire la moltiplicazione righe per colonne, con la generica matrice di trasformazione 3×3 .

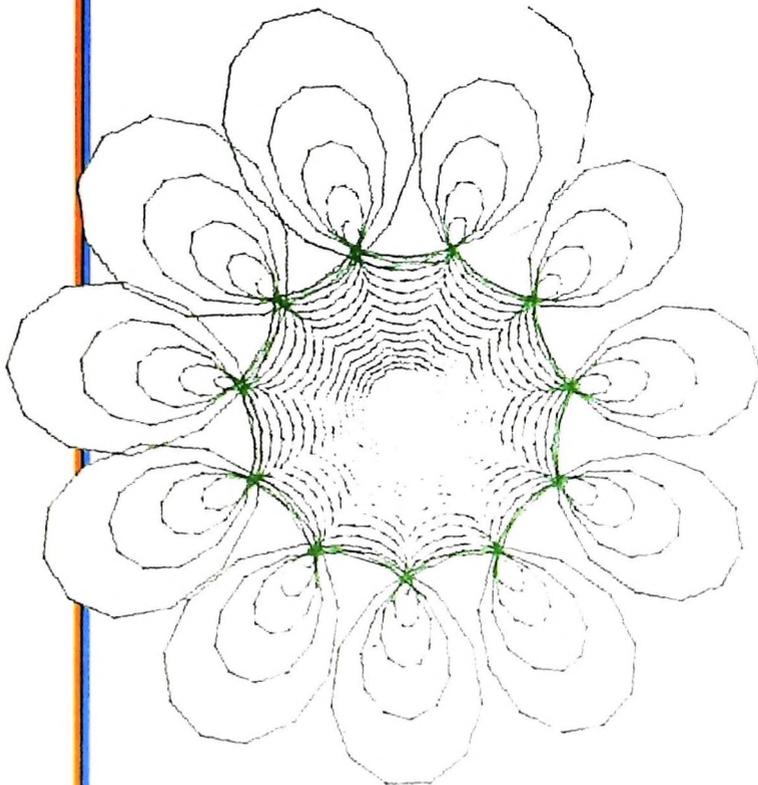
Ma cosa sono concettualmente le coordinate omogenee e che

Lo schema riproduce il sistema di coordinate omogenee grazie al cui utilizzo è possibile introdurre le tecniche di calcolo matriciale e trattare in maniera analitica anche i punti all'infinito.





Data 4,1,1,4,-1,5,1,4,1,5,-1,4,1,5,1,4,1,5,2,3



Data 1,3,2,-1,2,1,2,-1,-2,-1,5,5

```

10 DATA 1,0.3,2,-.65,2,.65,-2,.65,-2,.65,1,3.010
20 READ A1,A2,B1,B2,C1,C2,D1,D2,E1,E2,F1,F2
30 TMX=200
40 TINC=.1
50 CR=2
60 CLS
70 T=0
80 GOSUB 1000
82 INPUT "VIDEO (V) 0 PLOTTER (P)"; S$
84 IF S$ <> "V" AND S$ <> "P" THEN 82
86 IF S$="V" THEN 300
90 LPRINT CHR$(18)
100 LPRINT "C2"
110 A$="M"+STR$(X/CR+200)+","+STR$(Y/CR+10)
115 LPRINT A$
120 GOSUB 1000
130 A$="D"+STR$(X/CR+200)+","+STR$(Y/CR+10)
140 LPRINT A$
150 T=T+TINC
160 IF T<=TMX THEN 120
200 END
290 REM DEFINIZIONE DELLA WINDOW E
291 REM DELLA VIEWPORT PER USCITA SU VIDEO
300 W1=2 : W2=400
310 W3=-191 : W4=208
320 V1=80 : V2=143
330 V3=0 : V4=63
335 REM CALCOLO DELLE COSTANTI NELLE
336 REM TRASFORMAZIONI DI WINDOW
340 SX=(V2-V1)/(W2-W1)
350 SY=(V4-V3)/(W4-W3)
360 CLS
400 X0=X/CR+200 : Y0=Y/CR+10
410 X1=SX*(X0-W1)+V1 : Y1=SY*(Y0-W3)+V3
430 GOSUB 1000
450 GOSUB 2000
500 LINE (X1,Y1)-(X2,Y2)
510 X1=X2 : Y1=Y2
520 T=T+TINC
530 IF T<TMX THEN 430
540 END
1000 REM SUBROUTINE DI CALCOLO
1010 R1=100*EXP(-A1*.01*T)
1020 R2=100*EXP(-A2*.01*T)
1030 S1=R1*SIN(F1*T)
1040 S2=R2*SIN(F2*T)
1050 T1=R1*COS(F1*T)
1060 T2=R2*COS(F2*T)
1070 X=B1*S1+B2*S2+C1*T1+C2*T2
1080 Y=D1*S1+D2*S2+E1*T1+E2*T2
1090 RETURN
2000 REM SUBROUTINE PER IL CALCOLO
2001 REM DELLE COORDINATE VIDEO
2010 X0=X/CR+200 : Y0=Y/CR+10
2020 X2=SX*(X0-W1)+V1
2030 Y2=SY*(Y0-W3)+V3
2050 RETURN

```

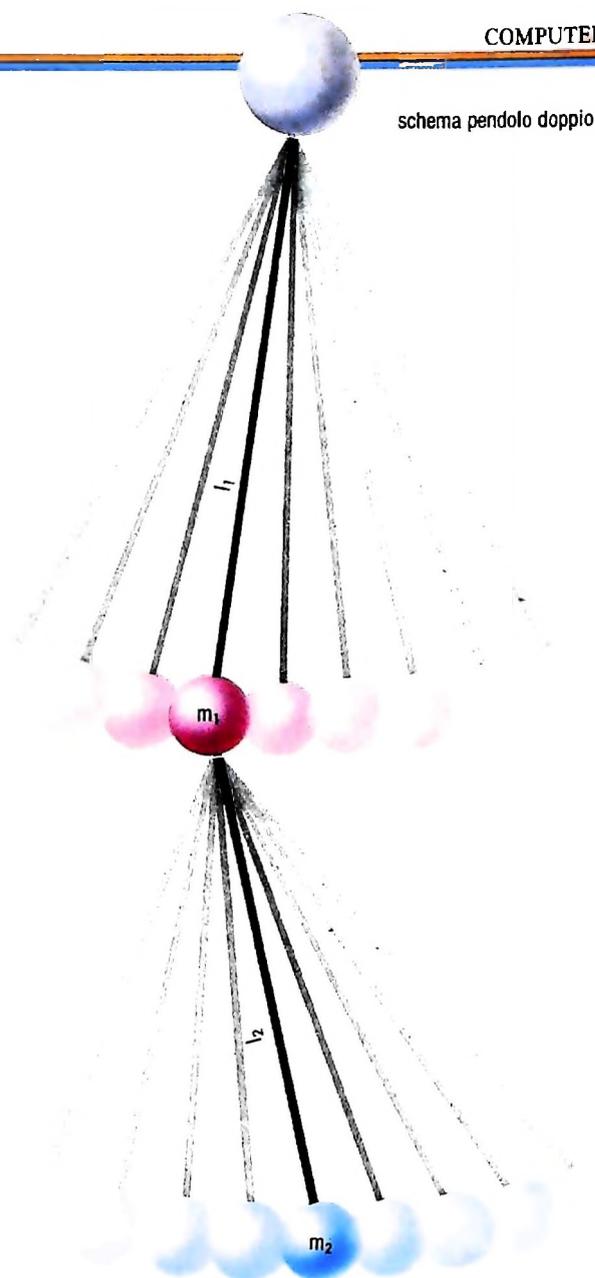
Divertimenti matematici

Abbiamo già avuto modo di vedere come alcune equazioni matematiche dia-no origine a soluzioni grafiche molto divertenti. Un valido esempio è costituito da questo programma, scaturito quasi spontaneamente da uno studio di meccanica razionale, che riproduce il "diagramma polare di un pendolo doppio smorzato". Per "pendolo doppio" si intendono due masse m_1 e m_2 fissate a due aste l_1 e l_2 nel modo illustrato in figura e che si muovono per il solo effetto del proprio peso. Quando di un tale sistema viene studiato il moto, deve essere considerato anche l'ambiente nel quale lo si considera immerso (per esempio aria, acqua, olio, liquidi diversi ecc.), che eserciterà un effetto di resistenza al moto stesso: questo è il significato del termine "smorzato". Per ogni situazione è infatti possibile stabilire un coefficiente che misuri il relativo grado di smorzamento. Infine, per "diagramma polare" si intende la rappresentazione grafica su un piano di un particolare tipo di moto del pendolo doppio: ossia un movimento polare è un movimento rigido durante il quale un solo punto detto *polo* rimane fisso (per esempio il punto in cui l'asta l_1 è vincolata a un sostegno). Esso è un movimento per cui lo spostamento è rotatorio, in cui l'asse di rotazione, pur variando nel tempo, passa sempre per il polo.

Ringraziamo qui l'ing. Luini del Politecnico di Milano che ci ha cortesemente segnalato l'algoritmo di calcolo.

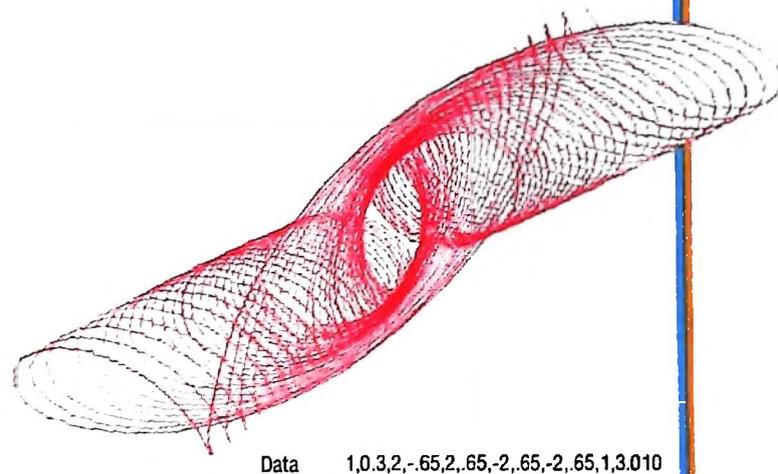
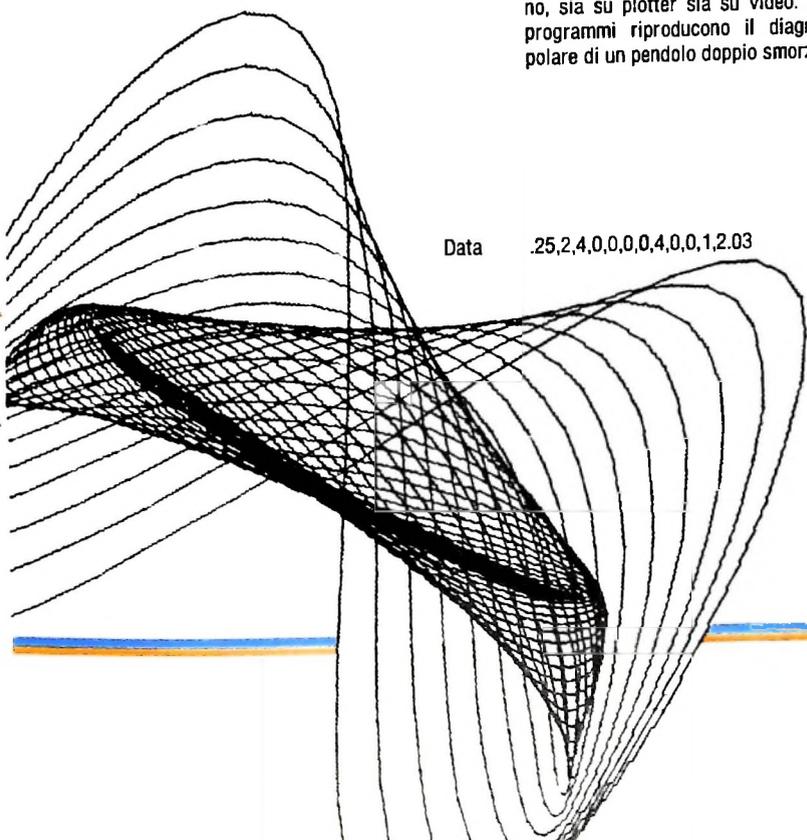
Il programma è stato stilato in maniera tale da poter offrire l'uscita sia su plotter che su video. Ovviamente, data la sua più alta risoluzione, il disegno su plotter risulterà più nitido e preciso.

I dati sui quali si effettuano i calcoli nella subroutine 1000, che vengono passati in input nella 10 e nella 20, costituiscono i parametri del sistema fisico in questione, ossia del pendolo doppio: essi sono tipicamente le lunghezze l_1 e l_2 dei pendoli, le masse m_1 ed m_2 , i coefficienti di smorzamento che agiscono sui due pendoli e le condizioni iniziali di velocità, di accelerazione e di angolazione. Non stiamo qui a dettagliare ulteriormente gli aspetti fisici del problema, ma segnaliamo semplicemente che variando il valore dei parametri dell'istruzione 10, si ottengono diagrammi differenti, dovuti al diverso comportamento che verrebbe ad assumere il sistema fisico simulato. Nelle immagini sono riprodotti quattro tipi di diagramma ottenibili da questo programma utilizzando la serie di dati specificata in ogni figura. Segnaliamo infine che l'esecuzione completa del disegno richiede almeno venti minuti.



schema pendolo doppio

Alcuni disegni ottenibili mediante la serie di dati specificati accanto a ciascuno, sia su plotter sia su video. Tutti i programmi riproducono il diagramma polare di un pendolo doppio smorzato.



significato geometrico assume l'artificio formale specificato? Per rispondere a questa domanda, bisogna aprire una breve parentesi di geometria analitica. Sappiamo che le coordinate ortogonali di un punto del piano sono le distanze del punto da una coppia di assi perpendicolari. Questo sistema, però, viene meno per i punti all'infinito del piano stesso, detti *punti impropri*, per i quali non è possibile assegnare valori numerici finiti alle coordinate ortogonali che dovrebbero individuarli. Quindi se vogliamo poter applicare i metodi analitici anche ai punti all'infinito, e questo potrebbe servirci per le proiezioni, è necessario trovare un sistema di coordinate che comprenda i punti impropri allo stesso modo dei *punti propri*, ossia quelli con coordinate di valore finito.

Il miglior modo per introdurre questo sistema di coordinate è quello di supporre il piano considerato, che indichiamo con la lettera P e le cui coordinate stabiliamo che siano X e Y , come situato nello spazio a tre dimensioni, nel quale siano state definite le coordinate ortogonali x, y, z . Poniamo il piano P parallelamente al piano coordinato x, y (figura di pagina precedente) e a distanza 1 al di sopra di esso, in modo che ogni suo punto abbia coordinate $(X, Y, 1)$. Prendendo l'origine O del sistema di coordinate come centro di proiezione, la geometria analitica ci dice che ogni punto Q del piano P determina un'unica retta passante per O e viceversa, ossia ogni retta passante per O interseca il piano P in un solo punto.

Definiamo ora per i punti del piano P un sistema di riferimento di "coordinate omogenee". Per trovare le coordinate omogenee di un punto proprio Q del piano P , prendiamo la retta passante per O e per Q e scegliamo su questa retta un punto qualsiasi T diverso da Q . Dalla geometria analitica sappiamo che le coordinate dei punti appartenenti ad una stessa retta passante per l'origine sono uguali fra loro a meno di un fattore moltiplicativo, per cui le coordinate spaziali ordinarie x, y, z di T si possono assumere come "coordinate omogenee" di Q . In particolare le coordinate $(X, Y, 1)$ di Q stesso costituiscono una terna di coordinate omogenee relative al punto Q . Quindi, ogni altra terna di numeri (tX, tY, t) , con t diverso da 0, rappresenterà ancora per il punto Q una terna di coordinate omogenee, poiché tutti i punti della retta QQ diversi da O avranno coordinate di questa forma (abbiamo escluso il punto $0,0,0$, perché appartiene a tutte le rette passanti per O e non può servire a distinguerle).

Questo sistema di coordinate richiede tre numeri invece di due per individuare la posizione di un punto nel piano, e presenta quindi lo svantaggio che le coordinate di un punto non sono determinate in modo unico, ma soltanto a meno di un fattore arbitrario t . Ora però, e questo è il grande vantaggio, anche i punti all'infinito del piano P sono compresi nella rappresentazione cartesiana: un punto Q all'infinito è determinato da una retta passante per O e parallela al piano P . Ogni punto T di questa retta avrà coordinate della forma $(x, y, 0)$. Quindi le coordinate omogenee di un punto all'infinito del piano P sono della forma $(X, Y, 0)$. Ricordiamo infatti che un piano e una retta a esso parallela si intersecano in un punto all'infinito.

Quindi possiamo affermare che la rappresentazione a tre dimensioni di un vettore bidimensionale, o in generale la rap-

presentazione di un vettore n -dimensionale mediante un vettore a $n+1$ dimensioni, è chiamata "rappresentazione in coordinate omogenee" e corrisponde, se si vuole, all'inverso della proiezione: ossia per passare dalla rappresentazione omogenea a quella cartesiana bisogna operare una proiezione (per esempio dallo spazio tridimensionale a quello bidimensionale).

In definitiva l'utilizzo delle coordinate omogenee ci consente di risolvere due problemi: semplificare le tecniche di calcolo matriciale, rendendo conformabile il vettore che rappresenta un punto con la matrice globale di trasformazione, e consentire di trattare in maniera analitica anche i punti all'infinito.

La matrice di trasformazione tridimensionale

Oltre alla necessità di rappresentare un oggetto tridimensionale, un'altra fondamentale esigenza della Computergrafica è quella di poter ruotare, traslare, scalare e proiettare l'oggetto stesso, al fine di rendere esattamente comprensibile la sua forma geometrica. Per fare tutto ciò con il computer dobbiamo estendere le nostre conoscenze sulle trasformazioni geometriche dal caso bidimensionale a quello tridimensionale. Utilizzando allora quanto abbiamo precedentemente visto, introduciamo le coordinate omogenee per questo tipo di ambientazione.

Un punto $[x \ y \ z]$ dello spazio tridimensionale è rappresentabile, per quanto visto prima, in forma omogenea dal vettore a 4 dimensioni $[x \ y \ z \ 1]$ o $[tx \ ty \ tz \ t]$. Noi, per comodità di calcolo, utilizzeremo sempre la prima notazione, se il punto non è improprio. La generica matrice di trasformazione, che nel caso bidimensionale era del tipo 3×3 , diventa ora, per la grafica a tre dimensioni, una matrice 4×4 , ossia del tipo:

$$T = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

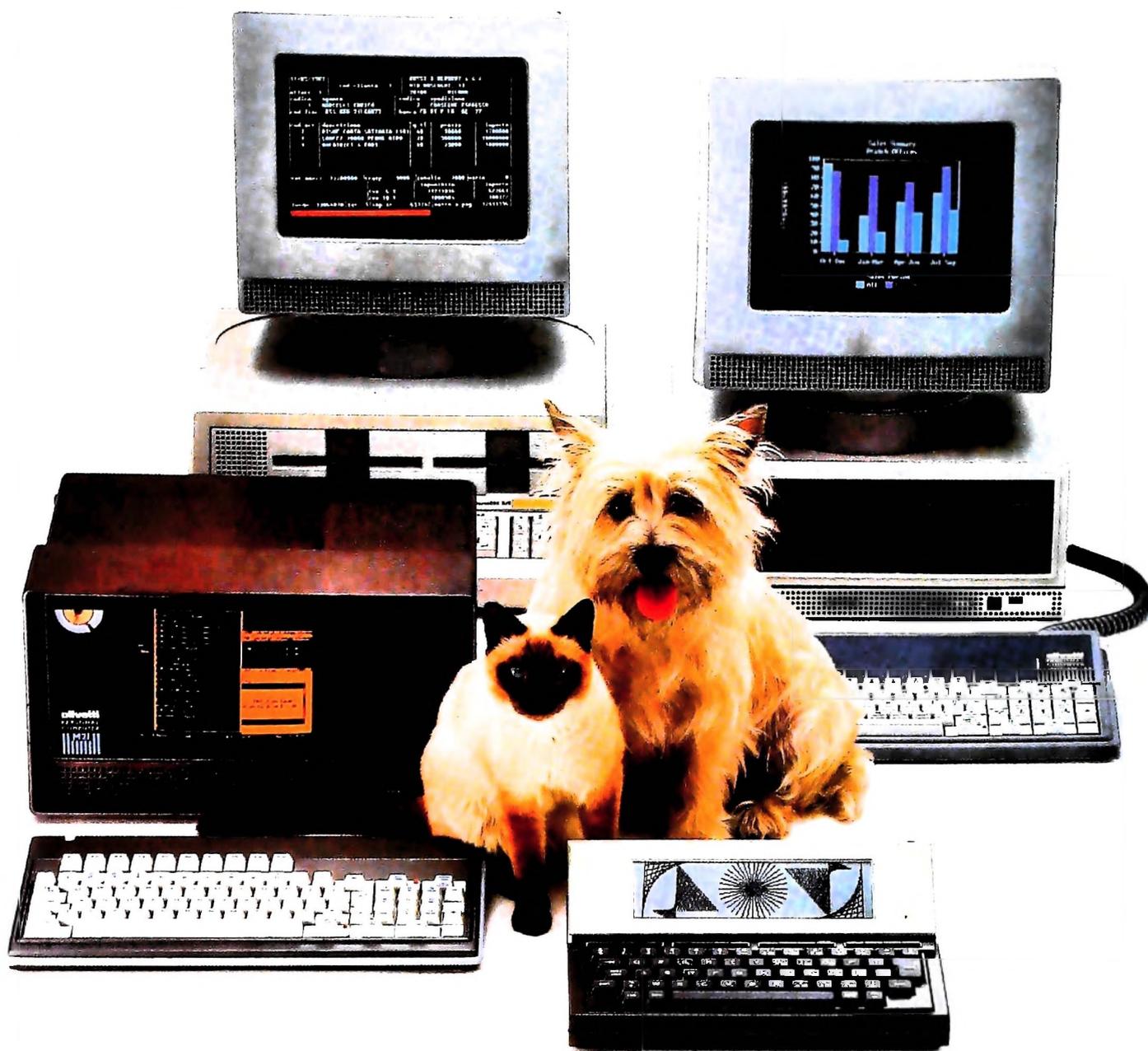
in cui i generici elementi a_{ij} possono assumere diversi valori numerici. Questa matrice può essere divisa in quattro sottomatrici nel seguente modo:

$$\begin{bmatrix} & & & \overline{3} \\ & & & x \\ 3 \times 3 & & & 1 \\ \hline & & & \\ 1 \times 3 & & & 1 \times 1 \end{bmatrix}$$

La matrice 3×3 produce le trasformazioni di scala e di rotazione; la matrice riga 1×3 genera le traslazioni mentre la matrice colonna 3×1 serve per le trasformazioni prospettiche. L'ultimo singolo elemento, indicato come matrice 1×1 , assumerà, nella nostra trattazione, sempre il valore 1.

Mediante la generica matrice quadrata 4×4 è quindi possibile combinare tutte le trasformazioni geometriche in un'unica entità matematica.

LA FAMIGLIA DEI PERSONAL COMPUTER OLIVETTI



FRIENDLY & COMPATIBLE

Questa famiglia di personal computer compatibili tra loro e con i più diffusi standard internazionali, non ha rivali per espandibilità e flessibilità. Prestazioni che su altri diventano opzionali, sui personal computer Olivetti sono di serie. Per esempio M24 offre uno schermo ad alta definizione grafica, ricco di 16 toni o di 16 colori e con una risoluzione di 600x400 pixel; mentre la sua unità base dispone di 7 slots di espansione, fatto questo che gli consente di accettare schede di espansione standard anche se utilizza un microprocessore a 16 bit reali (INTEL 8086). Ma ricchi vantaggi offrono anche tutti gli altri modelli.

Basti pensare che tutte le unità base includono sia l'interfaccia seriale che quella parallela. Oppure basti pensare all'ampia gamma di supporti magnetici: floppy da 360 a 720 KB o un'unità hard disk (incorporata o esterna) da 10 MB. La loro compatibilità, inoltre, fa sì che si possa far uso di una grande varietà di software disponibile sul mercato. Come, ad esempio, la libreria PCOS utilizzabile anche su M24. Come le librerie MS-DOS®, CP/M-86® e UCSD-P System®, utilizzabili sia da M20 che da M21 e M24.

MS-DOS è un marchio Microsoft Corporation
 CP/M-86 è un marchio Digital Research Inc.
 UCSD-P System è un marchio
 Regents of the University of California

olivetti

Per maggiori informazioni inviare il coupon a Olivetti
 Divisione Personal Computer, Via Meravigli, 12, 20123 Milano

NOVEMBRE
 INDIRIZZO
 CITTÀ
 TELEFONO

UN NUOVO MODO DI USARE LA BANCA.

CONSAULENZA



GLI INVESTIMENTI CON VOI E PER VOI DEL BANCO DI ROMA.

Il Banco di Roma non si limita a custodire i vostri risparmi. Vi aiuta anche a farli meglio fruttare. Come? Mettendovi a disposizione tecnici e analisti in grado di offrirvi una consulenza di prim'ordine e di consigliarvi le forme di investimento più giuste. Dai certificati di deposito ai titoli di stato, dalle obbligazioni alle azioni, il Banco di Roma vi propone professionalmente le varie opportunità del mercato finanziario. E grazie ai suoi "borsini", vi permette anche di seguire, su speciali video, l'andamento della Borsa minuto per minuto.

Se desiderate avvalervi di una gestione qualificata per investire sui più importanti mercati mobiliari del mondo, i fondi comuni del Banco di Roma, per titoli italiani ed esteri, vi garantiscono una ampia diversificazione.

Inoltre le nostre consociate Figeroma e Finroma forniscono consulenze per una gestione personalizzata del portafoglio e per ogni altra esigenza di carattere finanziario.

Veniteci a trovare, ci conosceremo meglio.

 **BANCO DI ROMA**
CONSCIAMOCI MEGLIO.