

# 60 CORSO PRATICO COL COMPUTER

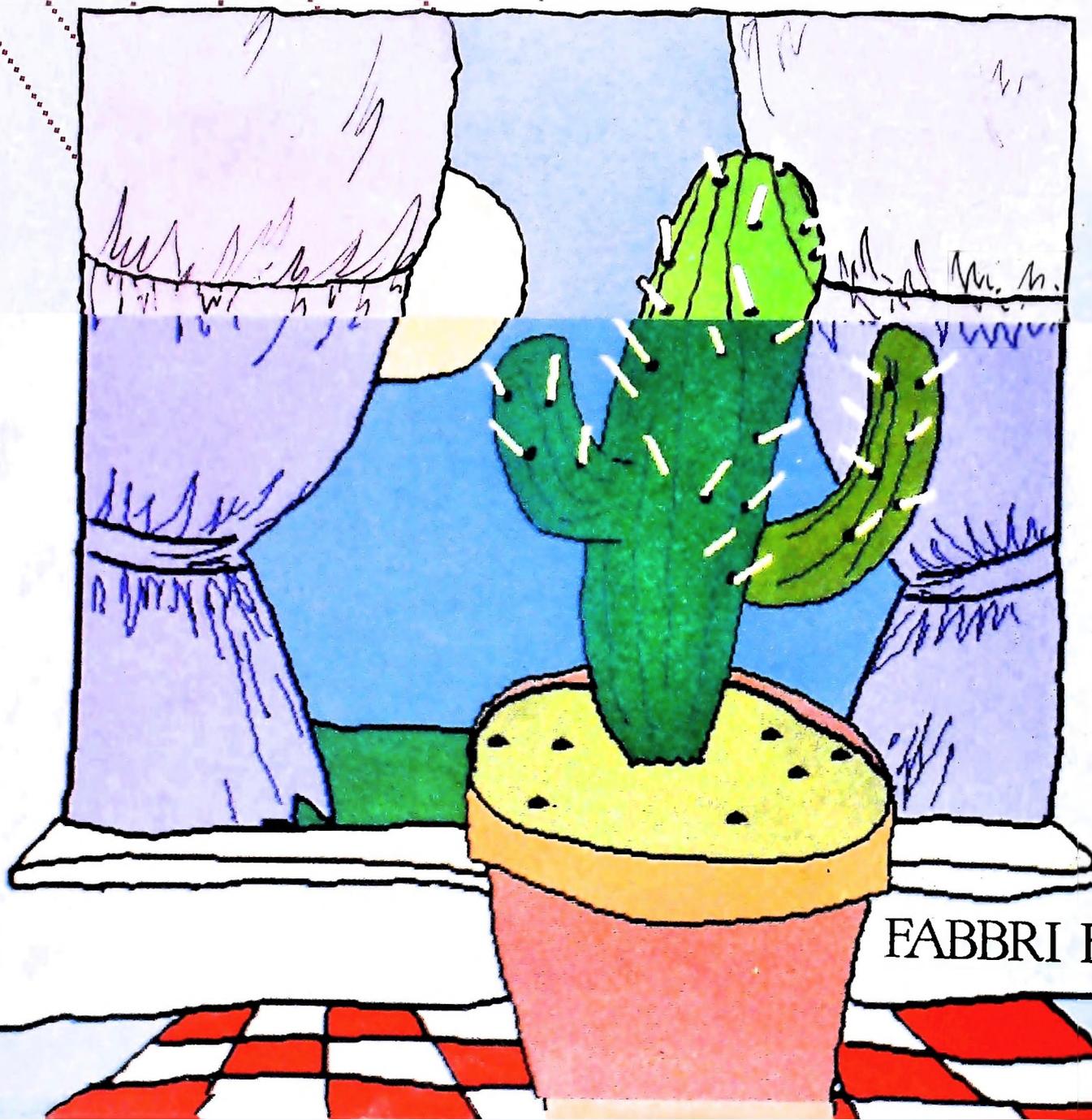
422121

è una iniziativa  
**FABBRI EDITORI**

in collaborazione con  
**BANCO DI ROMA**

e **OLIVETTI**

diretto da **GIANNI DEGLI ANTONI**



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 vetrofania 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  
Ricercatore universitario 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  
GIANLUIGI ROCCA, DANIELE MARINI, Eidos (TIZIANO BRUGETTI, BRUNO MOTTA), CLAUDIO PARMELLI, MAURIZIO OLIVA, MARCO VILLA, 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 FENGHI

Redazione  
CARLA VERGANI  
LOGICAL STUDIO COMMUNICATION

Art Director  
CESARE BARONI

Impaginazione  
BRUNO DE CHECCHI  
PAOLA ROZZA

Programmazione Editoriale  
ROSANNA ZERBARINI  
GIOVANNA BREGGE

Segretarie di Redazione  
RENATA FRIGOLI  
LUCIA MONTANARI

Corso Pratico col Computer - Copyright (©) sul fascicolo 1985 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Einaudi S.p.A., Milano - Copyright (©) sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sonzogno, Einaudi 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. 60 - 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.

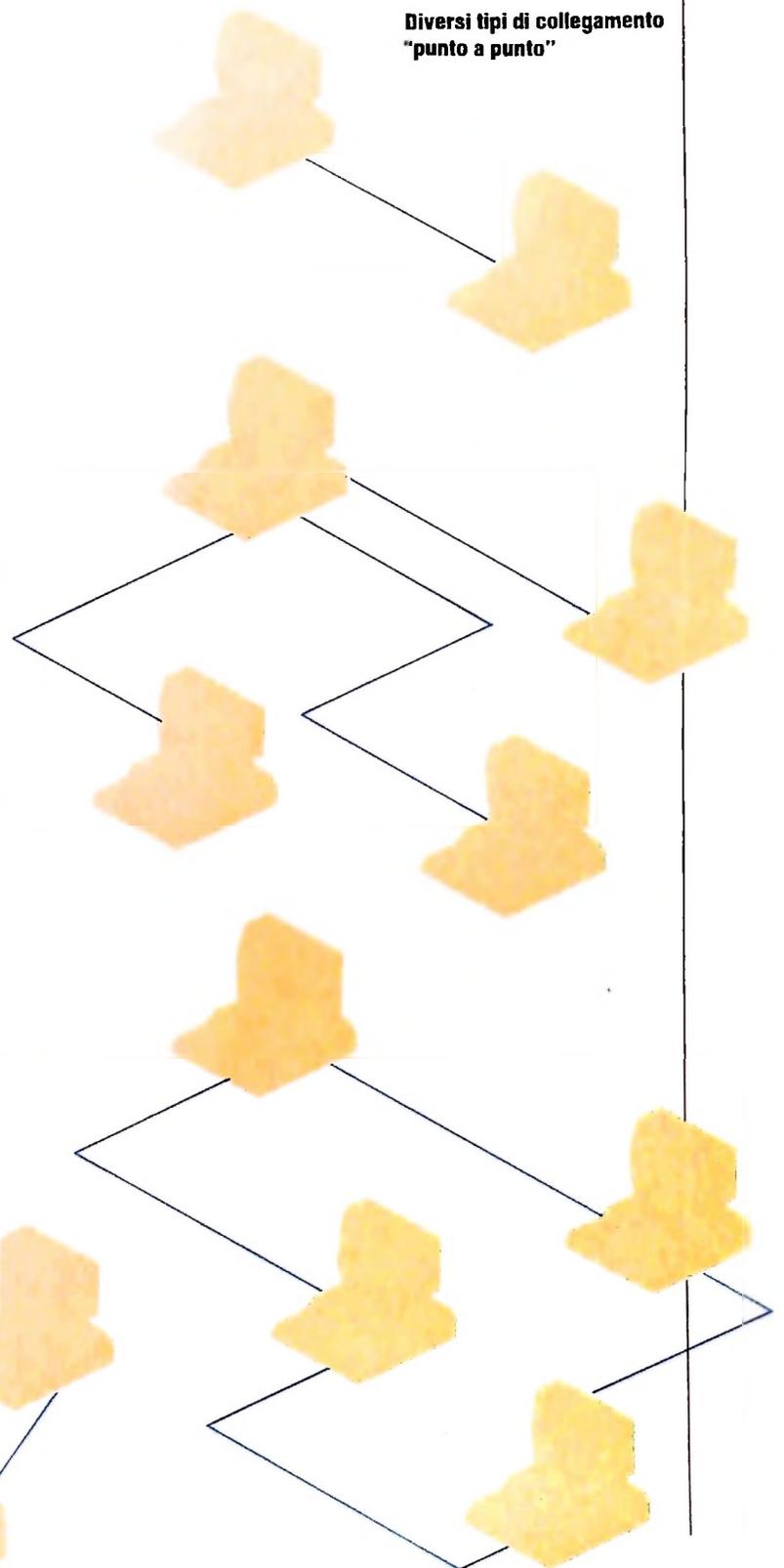
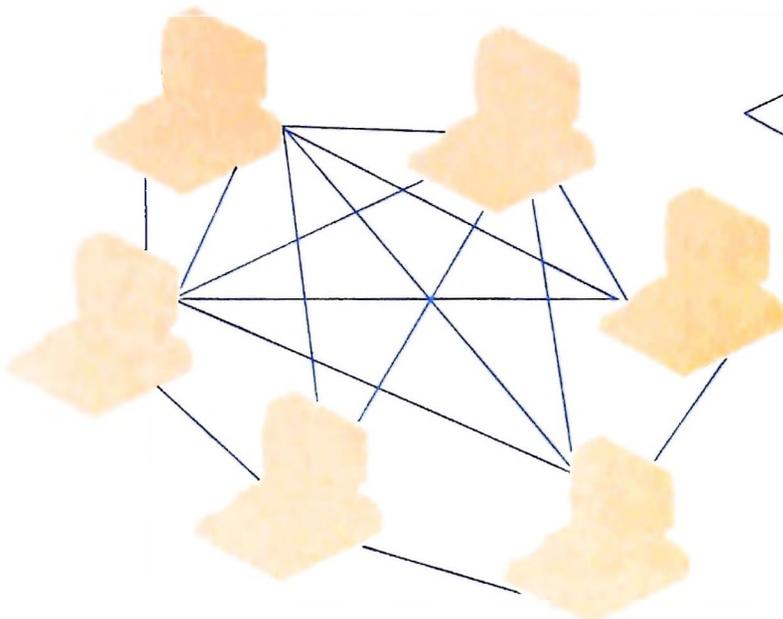
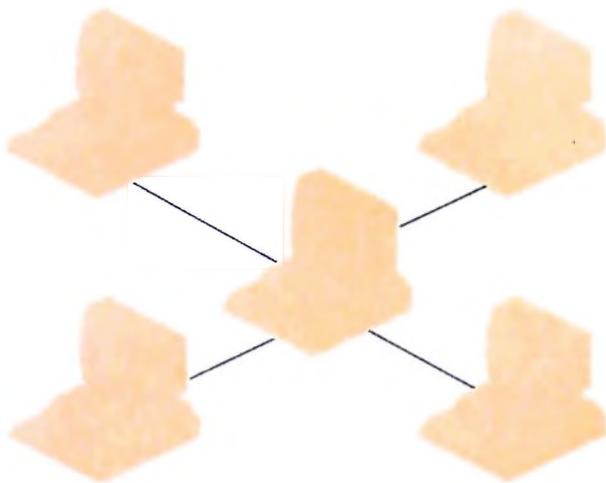
# LE RETI LOCALI (I)

I più comuni tipi di configurazione di reti e alcuni metodi per coordinare l'accesso al canale di comunicazione.

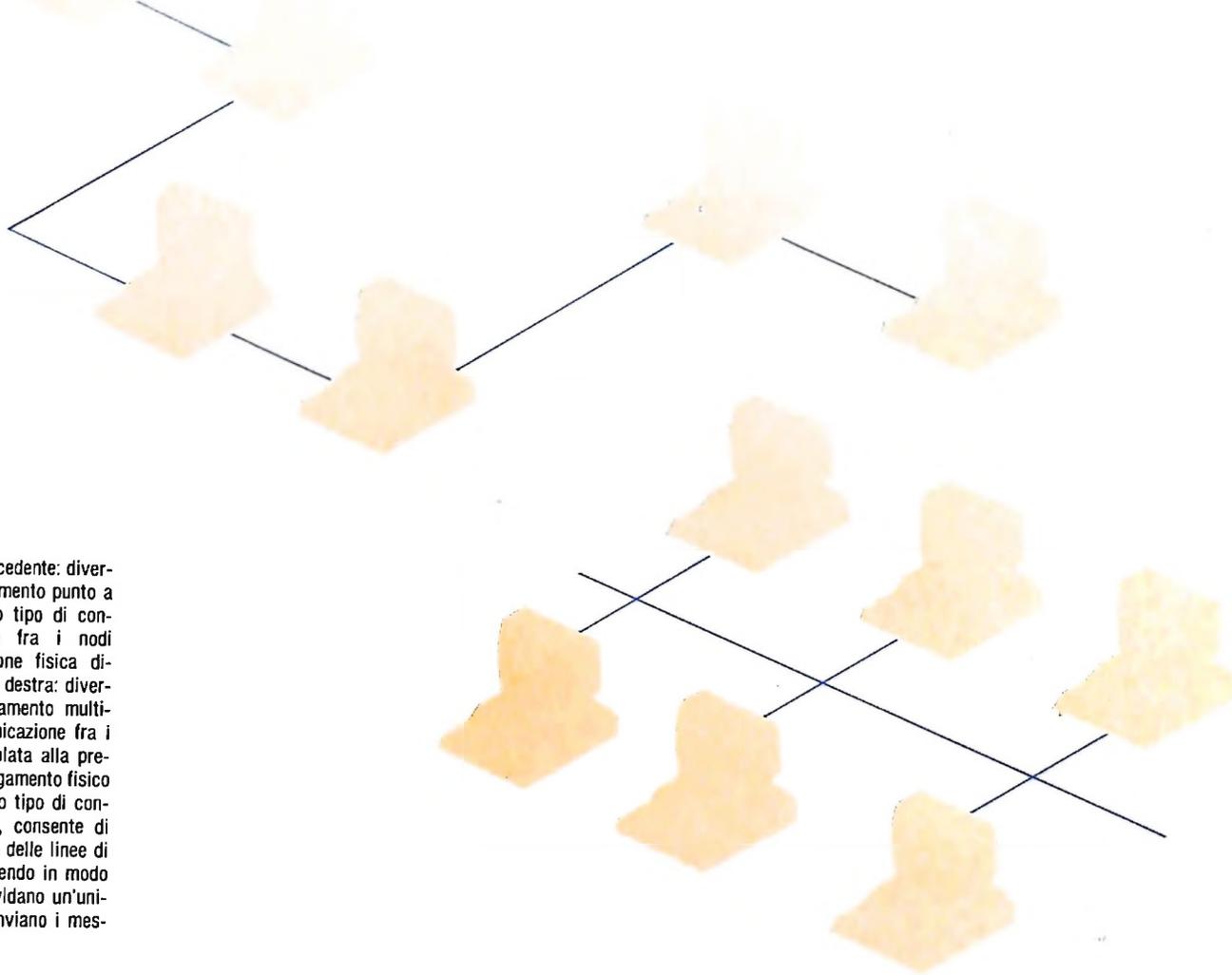
Lo sviluppo delle richieste software all'interno delle aziende ha portato, fin dai primi anni Settanta, a un grosso sovraccarico sulle strutture informatiche delle stesse, normalmente organizzate attorno a un grosso mainframe centrale con terminali distribuiti.

Le pressanti richieste dei dipartimenti periferici molto spesso sono state disattese e ciò ha portato allo sviluppo di una notevole autonomia in questo settore, manifestatasi con l'acquisto di minicomputer per soddisfare specifiche esigenze.

La conseguente esigenza di scambiare dati con il mainframe centrale ha portato alla definizione di connessioni in rete con



### Diversi tipi di collegamento "multipunto"



Nella pagina precedente: diversi tipi di collegamento punto a punto. In questo tipo di connessione esiste fra i nodi un'interconnessione fisica diretta. Sopra, e a destra: diversi tipi di collegamento multipunto. La comunicazione fra i nodi non è vincolata alla presenza di un collegamento fisico fra i nodi. Questo tipo di connessione, inoltre, consente di ridurre il numero delle linee di collegamento facendo in modo che i nodi condividano un'unica linea su cui inviano i messaggi.

lo stesso e si è quindi rivelata l'esigenza di compatibilità fra i protocolli di comunicazione delle apparecchiature e dunque per la rete.

La stessa situazione si è ripetuta di recente all'interno degli stessi dipartimenti con l'acquisizione di personal computer, impiegati per compiti specifici, che dovevano poi essere a loro volta collegati mediante una "rete locale" con il microcomputer del dipartimento.

I personal computer collegati in rete tendono inoltre a sostituire il minicomputer in tutti quei casi in cui si avverte l'esigenza di una nuova informatizzazione di un dipartimento o di una sede locale proprio per la loro maggiore agilità di funzionamento e per il loro minore costo.

Le reti locali connettono dunque sistemi di tipo diverso, solitamente abbracciano aree di pochi Km, sono contenute all'interno di edifici e hanno una velocità di trasmissione che va da 1 Mb (MegaBit) a 10 Mb al secondo.

Alla rete non sono collegati solo terminali o computer ma anche stampanti, dischi, nastri ed altre unità periferiche che vengono così condivise tra diversi utenti.

Il mercato delle reti è un settore in piena evoluzione ed esiste una agguerrita concorrenza fra le società produttrici di reti e di software di comunicazione. In questo articolo si vedranno sia le architetture di rete che gli standard esistenti a livello di

protocolli di comunicazioni così come alcuni esempi di reti presenti sul mercato.

### Le architetture di rete

Le architetture di rete definiscono la disposizione geometrica dei collegamenti e dei nodi che costituiscono la rete stessa, dove con *nodo* si intende un punto terminale della rete.

Esistono due tipi di connessioni per collegare dei nodi in una rete: *punto a punto* e *multipunto* di cui alcuni esempi sono mostrati nelle figure della pagina precedente.

Il tipo *punto a punto* connette due nodi senza passare per nessun nodo intermedio; in questo tipo di connessione esiste cioè un collegamento fisico fra i nodi e la comunicazione fra tutti i nodi della rete è possibile solo quando ognuno di questi sia collegato con tutti gli altri.

Diversamente il *multipunto* permette una struttura più semplice in quanto permette a ogni nodo di ritrasmettere, passare, un messaggio che non è a lui destinato, e quindi tutti i nodi connessi in una rete multipunto possono comunicare fra loro anche se fra gli stessi non esiste una interconnessione fisica diretta.

Il *multipunto* permette inoltre di ridurre il numero delle linee

di collegamento facendo in modo che i nodi condividano una unica linea su cui inviano i messaggi, come si vede nella figura di questa pagina.

Una configurazione di questo tipo per una rete è detta a *bus* ed è forse quella più comunemente usata nelle reti esistenti. Le reti a bus possono essere facilmente configurate ed adattate agli ambienti più diversi e sono particolarmente adatte a situazioni in cui esista una notevole variabilità nelle condizioni operative.

Questo tipo di rete può avere uno o due bus, su cui sono attaccate tutte le stazioni, e su cui verranno instradati i messaggi sia di andata che di ritorno nel primo caso o che verranno utilizzati quindi per i messaggi circolanti in una sola direzione nel secondo caso.

Un altro tipo di architettura abbastanza comune è quello a stella, mostrato nella figura qui sotto in cui tutti i nodi sono collegati a una unità centrale che si occupa di smistare o definire le comunicazioni fra i nodi o fra i nodi esterni e lo stesso

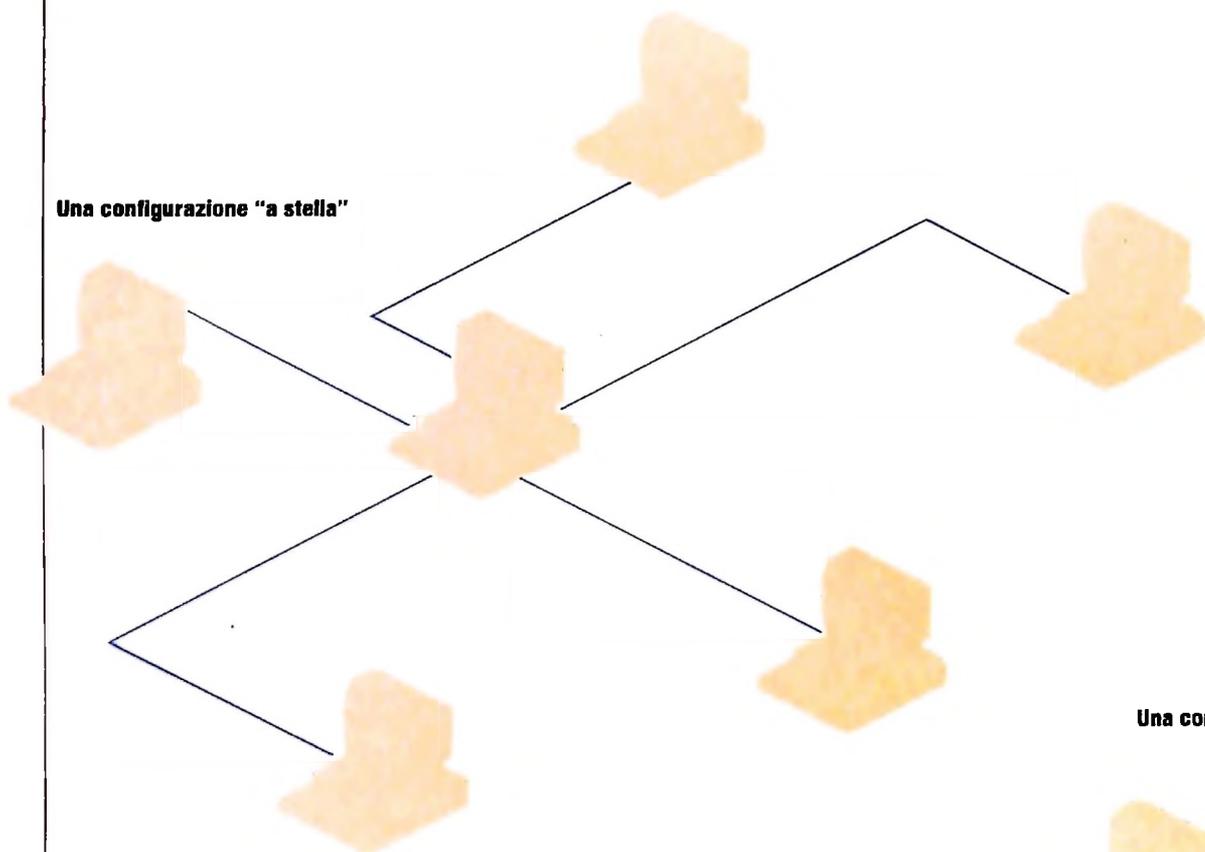
nodo principale. Il nodo di controllo può essere quello centrale, uno dei nodi esterni o in casi estremi la funzione di controllo può essere delegata a tutti i nodi.

Le reti a stella, come tutte quelle in cui esiste un solo nodo a cui è demandata una funzione di controllo o di smistamento, hanno però un punto critico, il nodo centrale, che può essere causa di problemi in quanto se questo si guasta l'intera rete cessa di funzionare e sono quindi necessarie delle misure di sicurezza e dei circuiti in sovrannumero per evitare lo spiacevole evento.

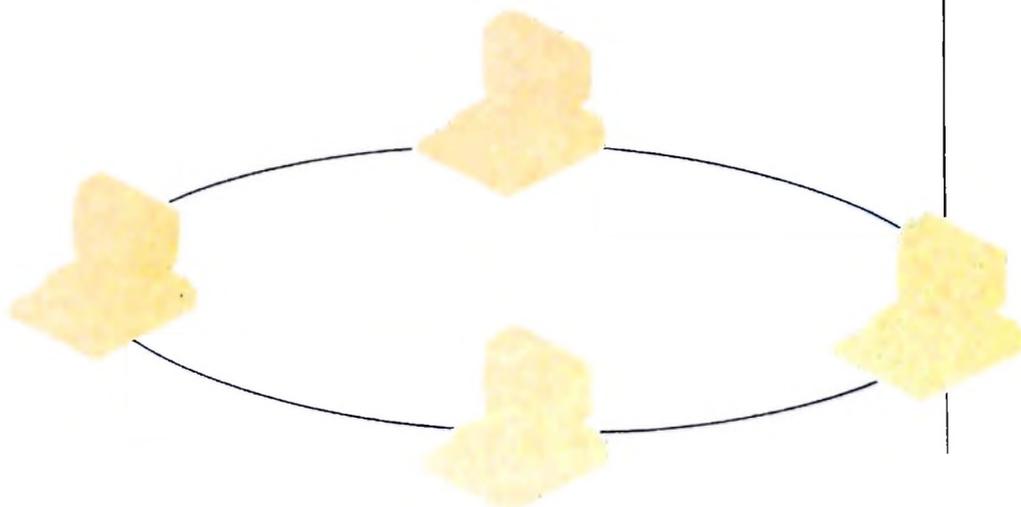
La capacità di trasmissione dell'intera rete è, per quanto concerne le configurazioni a stella, proporzionale alla potenza del nodo di smistamento centrale.

Un ultimo e diverso tipo di configurazione per una rete è quello ad anello, mostrato nella figura più in basso; in questo tipo di configurazione tutti i nodi, collegati in modo multipunto, sono disposti in modo da formare una configurazione circolare senza interruzioni.

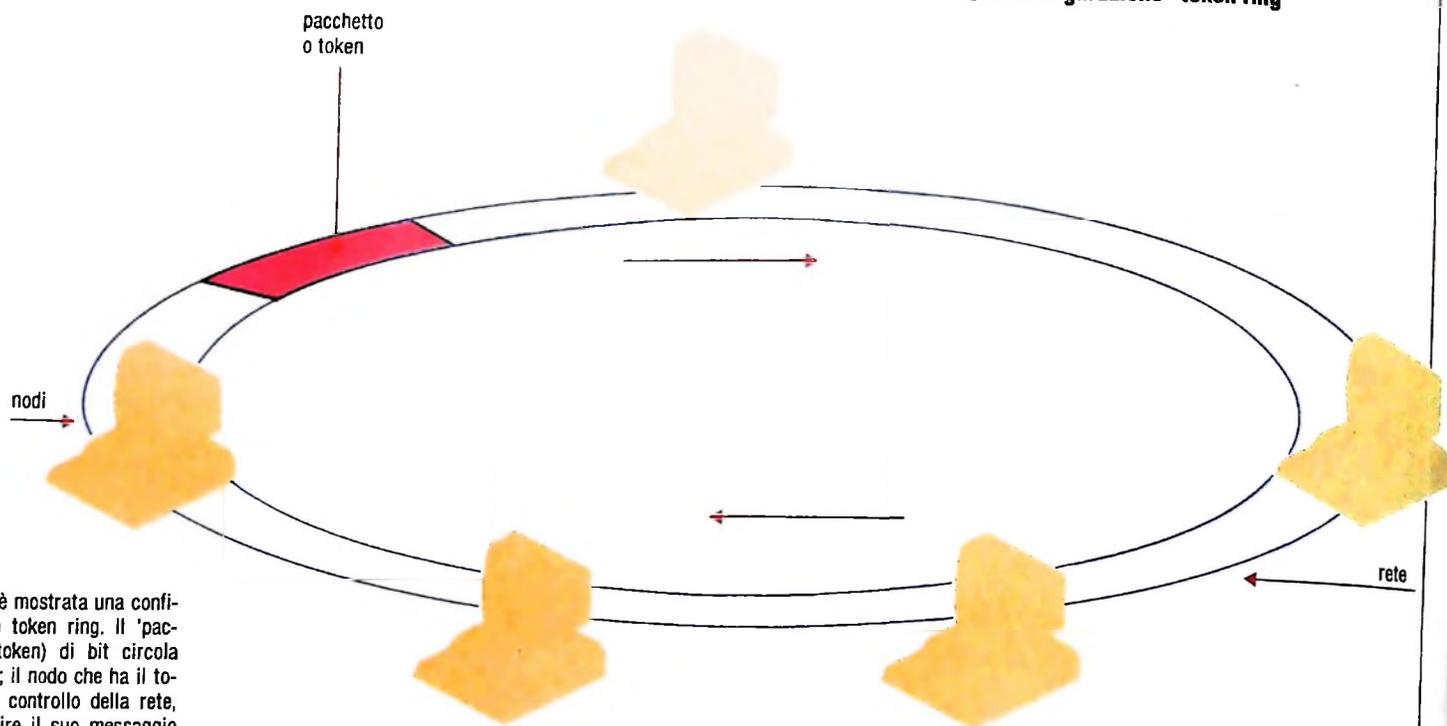
Una configurazione "a stella"



Una configurazione "ad anello"



Sopra: un altro tipo di disposizione geometrica dei collegamenti di rete è dato dalla configurazione a stella. A destra: rete ad anello.



In figura è mostrata una configurazione token ring. Il 'pacchetto' (token) di bit circola sulla rete; il nodo che ha il token ha il controllo della rete, può inserire il suo messaggio nel pacchetto e poi passarlo al nodo successivo.

I messaggi viaggiano sull'anello e ogni stazione deve essere in grado di riconoscere quelli diretti a se stessa o di inoltrare il messaggio, comportandosi come un ripetitore se il messaggio è diretto ad un'altra stazione.

Gli anelli forniscono un canale comune su cui tutti i nodi sono collegati in maniera logica. Se la logica della rete impone un controllo centralizzato, cioè che tutti i messaggi passino per un determinato nodo, la configurazione ad anello si riduce ad una configurazione a stella.

### Il controllo e l'accesso alla rete

Quando su di una rete sono presenti più nodi abilitati a trasmettere nasce il problema di definire quale di questi nodi abbia accesso alla rete, e sia quindi in grado di trasmettere, in un dato momento. Se più nodi iniziassero infatti a trasmettere contemporaneamente ci sarebbero degli evidenti problemi poiché i loro messaggi si sovrapporrebbero diventando così incomprensibili.

Per ovviare a questo inconveniente esistono più soluzioni: il più semplice consiste nel fare sì che esista un nodo principale che controlli in modo gerarchico l'accesso alla rete consentendo l'uso della stessa a un solo nodo alla volta, mediante l'assegnazione, per esempio, di un certo periodo di tempo a ogni nodo in sequenza.

Una soluzione di questo tipo, a scansione, comporta, come detto in precedenza, il pericolo di un guasto al nodo principale e della conseguente messa fuori uso di tutta la rete, cosa

che è vista come inaccettabile dai moderni progettisti di reti e inoltre non permette a un nodo di trasmettere, anche se il canale è libero, se non al suo turno, con una conseguente perdita di tempo e di efficienza.

### Tecniche a contesa

Una tecnica diversa e meno accentrante è quella a contesa che prevede gli eventuali conflitti o collisioni di messaggi a livello di progetto di rete e li sfrutta per assegnare l'accesso al canale di trasmissione.

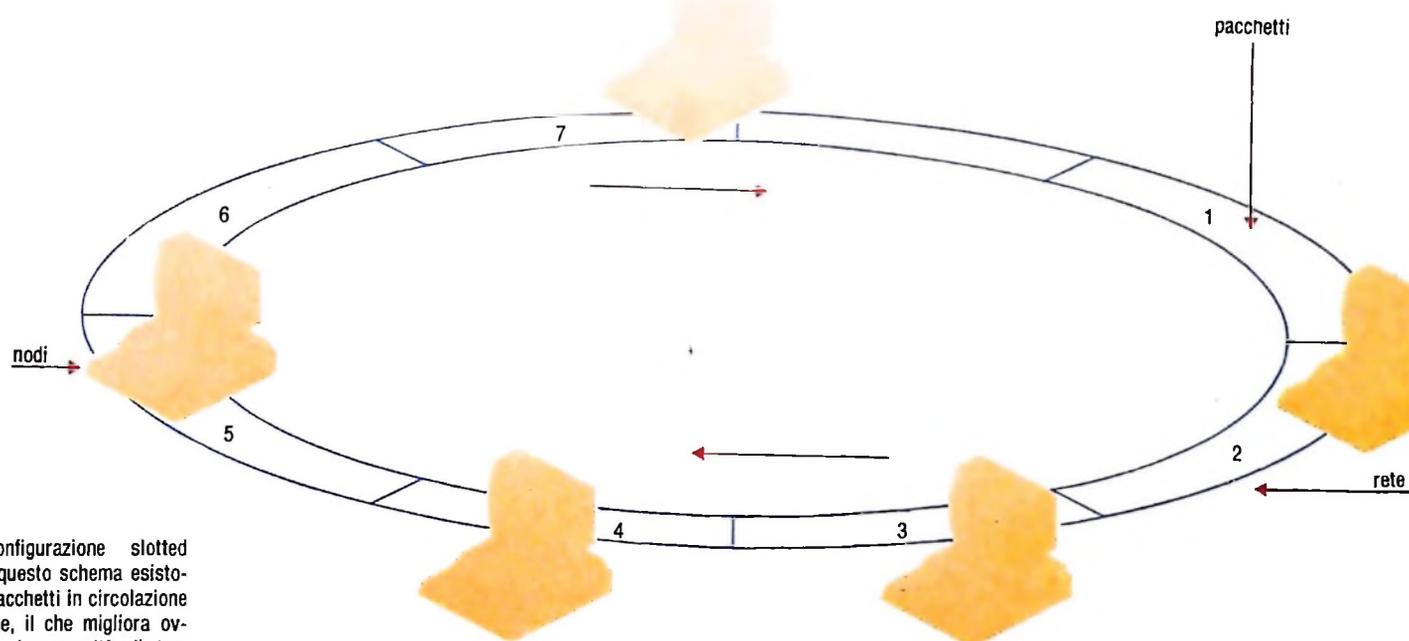
Una di queste tecniche, e forse la più usata, è la CSMA/CD (Carrier Sense Multiple Access with Collision Detection) che prevede che tutti i nodi abbiano la stessa possibilità di trasmettere un messaggio qualora il canale di comunicazione fosse disponibile, il che elimina per esempio il tempo di attesa di un nodo che si aveva nel caso precedente.

Ogni nodo della rete è in grado di rilevare la presenza di messaggi attraverso il Rilevamento di portante (Carrier Sense) ed è in grado quindi di rimandare la trasmissione del suo messaggio fino a quando il canale non è libero.

Due nodi potrebbero però trovare libero il canale e inviare il loro messaggio nello stesso istante, il che creerebbe una collisione. La collisione di due o più messaggi provoca però una variazione di segnale nel canale e quindi i nodi (Collision Detection) possono accorgersi di ciò ed attendere un certo periodo di tempo prima di reinviare il messaggio.

Il periodo di tempo che ogni nodo deve aspettare prima di

## Una configurazione "slotted ring"



Una configurazione slotted ring. In questo schema esistono più pacchetti in circolazione sulla rete, il che migliora ovviamente le capacità di trasporto delle informazioni da parte del sistema.

reinviare il proprio messaggio varia per ogni nodo poiché se fosse lo stesso per tutti si avrebbe una nuova collisione. Nel caso si verificasse una nuova collisione i nodi si rimettono in attesa questa volta per un tempo più lungo.

Nel CSMA/CD non esiste un ordine di accesso alla rete dei nodi e quindi non esiste un tempo massimo entro cui un determinato nodo può accedere alla rete, ma poiché ogni nodo controlla lo stato del canale si può definire un tempo statistico entro cui si ottiene l'accesso alla rete.

### Tecniche a token

Nelle configurazioni ad anello i dati trasmessi da un qualunque nodo effettuano una rotazione completa dell'anello prima di tornare al mittente, quindi i dati passano di stazione in stazione e vengono ritrasmessi.

Nei casi più semplici si usa, per coordinare l'accesso alla rete, un nodo principale che rileva l'utilizzo del canale e dà l'accesso allo stesso ad una sola stazione alla volta in modo, per esempio, sequenziale introducendo il concetto di gerarchia con i problemi definiti in precedenza.

La maggior parte delle reti ad anello, diversamente, adotta, per definire l'accesso al canale di comunicazione, uno schema *slotted ring* o uno schema *token ring*.

Il primo schema, definito dalla AT&T, suddivide l'anello della rete in pacchetti (speciali configurazioni di bit), mostrati nella figura della pagina precedente, che girano in continuazione e quando uno di questi si trova di fronte a un nodo

questo verifica se lo stesso è libero per inserirvi il proprio messaggio. Questo pacchetto circola poi sulla rete e ogni stazione vi può accedere per controllare se il messaggio è diretto alla stazione stessa, nel qual caso lo copia e inserisce un messaggio di ricevuto. Quando questo ripassa davanti al mittente il nodo lo rimuove liberando il pacchetto o eventualmente ritrasmettendolo se non è stato ricevuto.

La tecnica *token ring*, o *token passing* (adottata dall'IBM e dal MIT), mostrata nella figura in alto, prevede invece un solo pacchetto in circolazione sulla rete. Il nodo che ha il token (pacchetto), ha il controllo della rete e può inserire il suo messaggio nel token e quindi ripassarlo al nodo successivo che può a sua volta inserire un messaggio o ritrasmettere il token. Il fatto che un nodo possa trattenere il token solo per un certo periodo di tempo fa sì che il tempo massimo di attesa prima che un nodo possa trasmettere possa essere calcolato in modo molto semplice (è una funzione del numero di nodi) e permette quindi un approccio deterministico distinto da quello CSMA/CD che, come già detto, è essenzialmente probabilistico.

Lo schema *slotted ring* con più pacchetti in circolazione sulla rete permette un ottimo sfruttamento della stessa garantendo una eccezionale capacità di trasporto delle informazioni rispetto a qualsiasi altro sistema (CSMA.....). A dispetto di questa loro fondamentale caratteristica le reti *slotted ring* non hanno avuto un grande successo commerciale soprattutto a causa dell'elevata complessità dei protocolli di comunicazione che comporta interfacce software e hardware più costose di quelle usate in altri tipi di rete.

# POSTA ELETTRONICA (III)

Continuiamo la descrizione dei comandi e delle relative funzioni.

## DELETE

Si può abbreviare con le iniziali DE.

Il comando in oggetto cancella i messaggi o i file che non si desidera più tenere archiviati; tali messaggi sono salvati nel file di sistema DELETED, per permettere all'utente di recuperarli in caso di manovra errata o di ripensamenti.

Tutto ciò sarà possibile fino a quando non ci si scollegherà col comando LOGOUT.

In caso di caduta di linea o di time out, cioè in tutti i casi di scollegamento senza il comando LOGOUT, alla ripresa del collegamento, i messaggi cancellati saranno recuperabili, poiché archiviati nel file di sistema.

Il comando può assumere varie forme:

- 1) DELETE
- 2) DELETE n (per esempio: DELETE 8)
- 3) DELETE n<sub>1</sub>, n<sub>2</sub> (per esempio: DELETE 1,6)
- 4) DELETE n<sub>1</sub>, - n<sub>2</sub> (per esempio: DELETE 5-7)
- 5) DELETE ALL
- 6) DELETE nome-file (per esempio: DELETE TRANSITO)
- 7) DELETE, nome-file<sub>1</sub>, nome-file<sub>2</sub> (per esempio: DELETE TRANSITO, TEMP)

DELETE viene usato per cancellare il messaggio.

I formati 2,3,4,5 vengono utilizzati dopo un comando SCAN o SCAN più nome file; nel secondo esempio viene cancellato l'ottavo messaggio analizzato; nel terzo esempio vengono cancellati il primo e il sesto; nel quarto esempio i messaggi dal quinto al settimo, e infine nel quinto esempio vengono cancellati tutti i messaggi contenuti nell'archivio, escludendo il nome dell'archivio.

I formati 6,7 vengono utilizzati per cancellare uno o più file con tutti i messaggi in essi contenuti.

## DISPLAY

Si può abbreviare con le iniziali DI.

Il comando DISPLAY presenta, in funzione dell'argomento indicato, diverse informazioni riguardanti gli archivi del sistema e dell'utente, la lista degli utenti di Infomark e i dati statistici relativi all'utente sull'uso del sistema, compresi quelli relativi alla tariffazione.

Consideriamo per esempio i seguenti formati:

- 1) DISPLAY FILE
- 2) DISPLAY LIST
- 3) DISPLAY STATISTIC

DISPLAY FILE — Elenca i file di sistema e dell'utente con accanto il numero dei messaggi in essi contenuti.

DISPLAY LIST — Elenca gli utenti di Infomark sia come liste di distribuzione e sia, fornendo la lista, gli utenti in essa inclusi.

DISPLAY STATISTIC — Evidenzia il nome dell'utente, il relativo numero di accounting, il numero di collegamenti avuti col sistema, la quantità dei messaggi inviati, la quantità dei messaggi ricevuti, il tempo di effettiva connessione col sistema dall'ultimo scarico di accounting, la quantità dei messaggi archiviati, il numero dei file.

## EDIT

Si può abbreviare con la lettera E.

Il comando permette di modificare i campi della testata di un messaggio, oppure di ampliare, ridurre e modificare il testo di un messaggio che non sia stato né inviato né ricevuto.

Quando si è terminato di lavorare in EDIT, per riposizionarsi su COMMAND è necessario battere un punto (.) come primo carattere di una riga, e premere il tasto RETURN.

I campi FROM e DATE non si possono modificare, essendo riempiti automaticamente dal sistema al momento dell'invio del messaggio.

Una volta battuto il comando EDIT, viene richiesto se vi sono modifiche da apportare all'intestazione del messaggio o al testo vero e proprio, a meno che non si sia battuto direttamente il comando con accanto il campo da modificare: EDIT TEXT o HEADER, o CC, o TO, o SUBJECT, nel qual caso ci si troverà già posizionati sul campo da modificare.

Con il comando EDIT si possono:

1) aggiungere parole o frasi, sia in testa che alla fine del messaggio, ovvero inserire parole o frasi tra due parole esistenti nel messaggio;

2) cancellare parole o frasi del messaggio e quindi sostituirle, nella stessa posizione, con altre;

3) cancellare parole o frasi del messaggio.

Tutte le operazioni da fare per modificare il testo iniziano con un punto (.) come primo carattere di riga, seguito dal comando che occorre dare.

I sottocomandi dell'EDIT che permettono di localizzare e apportare modifiche al testo sono:

- 1) .LOCATE testo
- 1a) .LOCATE testo...testo
- 1b) .LOCATE ....testo
- 1c) .LOCATE testo....

- 1d) .LOCATE .TOP
- 1e) .LOCATE .BOTTOM
- 2) .DISPLAY
- 2a) .DISPLAY .ALL
- 3) .ERASE

**LOCATE** - Serve per localizzare, nel testo del messaggio, una particolare parola:

- 1a) da una parola all'altra parola
- 1b) dall'inizio del testo, sino alla parola ricercata
- 1c) dalla parola ricercata, sino alla fine del messaggio
- 1d) per posizionarsi all'inizio del messaggio
- 1e) per posizionarsi alla fine del messaggio

Localizzata la parola, il sistema visualizza **TFXT LOCATED**, oppure **TEXT NOT LOCATED**.

**.DISPLAY** - Serve per visualizzare il testo localizzato dal precedente comando **.LOCATE**

Il testo localizzato e visualizzato è racchiuso tra parentesi <.....>.

**.ERASE** - Cancella tutto ciò che è stato localizzato precedentemente e che viene visualizzato tra parentesi nel comando **.DISPLAY**.

Se si volesse modificare un messaggio ormai inviato o ricevuto, il sistema visualizzerà: "YOU CANNOT EDIT A MESSAGE THAT HAS BEEN SENT" (non si può editare un messaggio già inviato).

#### FILE

Si può abbreviare con le iniziali **FI**.

Il comando **FILE** archivia il o i messaggi in uno o più file.

Si possono creare tutti gli archivi che occorrono e chiamarli come si desidera, tenendo conto che il nome dell'archivio ha le seguenti limitazioni:

- deve iniziare con un carattere alfabetico
- deve avere lunghezza massima di 16 caratteri
- non deve contenere spazi
- non deve essere un comando Infomark o una parola speciale come **ALL,REST,IN,TO**.

Se il file è già stato creato, il sistema aggiungerà messaggi a messaggi, in caso contrario Infomark segnalerà che non esiste un file con tale nome (**YOU DON'T HAVE A.....FILE**) e chiederà se debba o meno crearlo visualizzando: "DO YOU WISH TO CREATE ONE? (Y or N):..."

È da notare che solo una copia del messaggio verrà archiviata, indipendentemente da quanti archivi la contengono.

Per archiviare un messaggio è sufficiente dare il comando **FILE** più il nome del file desiderato; l'**IN** è facoltativo.

Per esempio i comandi:

- 1) **FILE IN TEMP**
- 2) **FILE IN TEMP,PROVV**
- 3) **FILE 2,5 IN TEMP**
- 4) **FILE 2,5-7 IN TEMP**
- 5) **FILE 5,3,1 IN TEMP**
- 6) **FILE 5, REST IN TEMP**
- 7) **FILE ALL IN TEMP**

hanno il seguente significato che verrà ora descritto.

I comandi 1-2 si possono usare dopo un comando **READ** o **COMPOSE** per archiviare il messaggio in uno o più file.

I comandi 3-4 si possono usare dopo un comando **SCAN** o **SCAN** più nome file. Nel terzo esempio verranno archiviati il secondo e il quarto messaggio, nel quarto esempio verranno archiviati il secondo e dal quinto al settimo messaggio.

Negli esempi 5-6 si nota che il comando **FILE** può essere anche usato per riordinare i messaggi in un archivio. Nei suddetti esempi, dopo aver usato il comando **SCAN** o **SCAN** più nome file, i tre messaggi si archiveranno in ordine diverso nel file **TEMP**.

Il parametro **ALL** serve per copiare tutti i messaggi da un file all'altro dopo un comando di **SCAN** o **SCAN** più nome file.

#### FORWARD

Si abbrevia con le iniziali **FO**.

Il comando **FORWARD** permette di inoltrare a uno o più utenti un messaggio ricevuto, aggiungendo anche un commento o un altro messaggio.

Il comando **FORWARD** come il comando **COMPOSE**, richiede di riempire i campi dell'intestazione e indi il testo.

Per uscire dal comando e ritornare di nuovo su "COMMAND:" è necessario, alla fine della compilazione della testata, premere **RETUNR**, un punto come primo carattere della nuova riga, ed infine di nuovo **RETUNR**.

Dopodichè il sistema risponde "MESSAGE ENCLOSED".

Si può inoltrare un messaggio quante volte si vuole, ma anche non modificare i messaggi ricevuti o inviati.

Affinchè il messaggio accluso a quello ricevuto o inviato venga rinviato, è necessario il comando **SEND**.

Quando il destinatario riceve il messaggio, gli viene visualizzato prima il messaggio accluso e poi quello o quelli inviati (**FORWARDED MESSAGE**).

#### HELP

Si può abbreviare con **H**.

Il comando descrive la funzione ed i parametri validi del comando in oggetto.

#### LOGOUT

Porre fine alla sessione di lavoro.

Dopo questo comando è necessario riprendere dall'inizio la procedura di connessione.

Si ricorda che, prima di dare il comando **LOGOUT**, si possono recuperare i messaggi che sono stati cancellati nella corrente sessione di lavoro poiché essi sono archiviati nel file di sistema **DELETED**.

Prima di terminare definitivamente la sessione di lavoro, se si hanno dei messaggi nel file, **UNREAD**, il sistema li quantifica e chiede se si desidera leggerli prima di essere scollegati.

Ricevendo il sistema risposta affermativa (cioè battendo **Y**) la sessione in corso continua, visualizzando di nuovo "COMMAND": e potendo, così, leggere i messaggi non letti o usare gli altri comandi di Infomark.

In caso di risposta negativa (cioè battendo **N** o **RETURN**), nonostante si abbiano ancora dei messaggi da leggere il sistema chiude la sessione in corso.

# IL SISTEMA OPERATIVO UNIX (I)

Un sistema che, in tempi recenti, ha avuto una grande diffusione per la semplicità e l'efficacia con cui risponde a differenti e molteplici esigenze.

Nell'evoluzione dei sistemi operativi si è evidenziata nell'ultimo decennio la tendenza a produrre sistemi che, rispondendo semplicemente ed efficacemente a differenti e molteplici esigenze, potessero essere presenti su una vasta gamma di calcolatori, soprattutto mini e micro, tali cioè da costituire degli 'standard' per i potenziali utenti. Un contributo, in questo senso, è venuto senz'altro da Unix. L'analisi della sua storia, prima, della sua composizione e delle caratteristiche nonché degli aspetti generali di filosofia, poi, serviranno a farci comprendere i motivi della grande diffusione che questo sistema ha avuto in tempi recenti.

## La storia di Unix

L'idea di sviluppare un nuovo sistema operativo come Unix nacque dalla situazione di disagio in cui si trovava ad operare, attorno alla fine degli anni Sessanta, il gruppo di sviluppo software dei Laboratori Bell dove già spiccava la figura di Ken Thompson che diverrà poi famosissimo come maggiore artefice della nascita di Unix.

L'ostacolo principale per l'attività del gruppo Bell era questo: i tradizionali sistemi operativi non erano sufficientemente flessibili per supportare con facilità e versatilità i problemi tipici che nascono in un ambiente di sviluppo software.

L'obiettivo da raggiungere era un sistema operativo che si configurasse come reale supporto alla programmazione (qualcosa che, in anni successivi, sarà denominato *software factory*).

Unix è nato così come un sistema *general purpose*, con l'obiettivo minimale di proporsi come supporto per lo *sviluppo dei programmi*.

Com'era nelle intenzioni di Thompson, Unix doveva eccellere, nel suo peculiare campo di applicazione, per facilità, potenza, eleganza e versatilità. Queste erano le premesse storiche di Unix; vediamo ora l'evoluzione implementativa.

Ottenuto il benessere dei Bell Labs, Thompson cominciò a sviluppare una prima versione di Unix sui calcolatori PDP-7 e PDP-9. Scritto interamente in *Assembler*, in versione monoutente, si basava su alcuni concetti tratti da un sistema operativo, *Multics*, ideato per grandi calcolatori (il nome Unix trae origine proprio da questo).

Nel 1969 la prima versione era operativa e, raccolti i primi consensi, si decise di sviluppare una seconda versione multiutente per i più potenti calcolatori PDP-11.

Per questa occasione si volle utilizzare estesamente il C, un nuovo linguaggio di programmazione creato appositamente da Dennis M. Ritchie, anch'egli ricercatore dei Bell Labs. Questa scelta, se da un lato comportò una crescita del sistema operativo in termini di occupazione di memoria, gettò le basi per la sua diffusione sul mercato poiché Unix divenne più facilmente trasferibile su altri calcolatori.

La nuova versione vide la luce nel febbraio 1971.

A questo punto le qualità intrinseche di Unix, i suoi aspetti innovativi, uniti alla facilità di trasporto su altri calcolatori ne permisero una velocissima diffusione nelle università e nei laboratori di ricerca: tutto ciò consentì di arricchire e di affinare Unix anche dal punto di vista della interfaccia utente.

Si è giunti così, per passi successivi, ma sempre più rapidi, dalla versione 6 dei Laboratori Bell alla versione 7, con l'impronta determinante del gruppo di ricerca dell'Università di Berkeley.

Sensibilizzate dal successo che ormai Unix raccoglieva negli ambienti di studio e ricerca più avanzati negli anni Settanta, anche le case costruttrici di hardware cominciarono a credere al 'fenomeno Unix' e a intravedere un valido business per cui si progettaron piccoli e medi calcolatori che supportassero e ponessero in risalto le caratteristiche di Unix.

È impossibile riportare tutte le versioni e le modifiche, anche sostanziali, che Unix ha subito dalla sua nascita a oggi.

Vale la pena di ricordare che nella storia di Unix hanno avuto un peso determinante anche alcune software-house come la Microsoft e la Interactive che hanno arricchito Unix e lo hanno adattato a macchine diverse.

Sono proliferati anche sistemi operativi cosiddetti Unix-like che, partendo dai concetti originali di Thompson, si adattano a particolari campi di applicazione.

Unix vanta ormai migliaia di installazioni su mini e microcalcolatori e si trova nelle sue più svariate vesti applicato con successo anche nelle applicazioni industriali.

Secondo gli esperti, è prevedibile che Unix subisca una diffusione di massa nei prossimi anni, configurandosi come un sistema operativo "standard" per calcolatori della fascia mini-micro: è sulla base di queste previsioni che la maggior parte

## Lezione 59

**I comandi per guidare il formato di stampa**

Nella precedente lezione abbiamo evidenziato cosa succede quando si vuole fare una stampa che presenti una certa "eleganza" se molti dati sono coinvolti.

In particolare abbiamo visto quali difficoltà sorgano nel governare la stampa con il solo uso dei separatori "," e ";". I nostri problemi nascevano soprattutto dal fatto che tali separatori sono troppo rigidi. Il BASIC ci mette a disposizione alcune modalità di uso delle istruzioni PRINT e LPRINT che fanno però al caso nostro. Si tratta della istruzione PRINT USING, che specifica il "formato" con cui i dati devono essere visualizzati:

PRINT USING "stringa che descrive il formato"; lista di stampa

in essa:

- la stringa che descrive il formato può essere variamente composta, come vedremo e descrive come ciascuna variabile deve essere visualizzata nella lista;

- la lista di stampa è una normale lista di variabili o costanti, separate da "," o ";".

Riprendiamo il nostro esempio di stampa della tavola pitagorica: supponiamo di volere che ogni valore stampato occupi un campo di una certa lunghezza (espressa in termini di numero di caratteri), allineato sulla destra. Ciò è possibile usando il comando di formato

"f" "f" "f" "f"

ove il numero di caratteri "f" specifica la lunghezza del campo.

Sostituiamo a ogni LPRINT una LPRINT USING (ricordiamoci che tutti i discorsi che stiamo facendo si applicano altrettanto bene all'istruzione PRINT), ma, per semplicità, imponiamoci per ora che un comando di formato si applichi a tutti gli elementi della lista di stampa.

Osserviamo com'era fatto il programma visto nella lezione precedente:

```

10 REM Visualizza una tavola pitagorica
15 LPRINT "TAVOLA PITAGORICA"
20 INPUT "Quale dimensione";N
22 REM Visualizza prima riga
25 LPRINT "      I";
30 FOR I = 2 TO N
40 LPRINT I;"  ";
50 NEXT I
55 LPRINT
58 REM Visualizza linea di divisione
60 FOR I=1 TO N*6+6
70 LPRINT "-";
80 NEXT I
85 LPRINT
88 REM Calcola prodotti e li visualizza, riga
per riga
90 FOR I=2 TO N
95 LPRINT I;"  I";
100 FOR J=2 TO N
110 LPRINT I*J;"  ";

```

```

120 NEXT J
130 LPRINT
140 NEXT I

```

In esso le istruzioni 40, 95 e 110 stampano valori numerici e stringhe contemporaneamente. Ricostruiamolo disaccoppiando le stampe di valori numerici e di stringhe con più istruzioni LPRINT, e usando la forma USING:

```

10 REM Visualizza una tavola pitagorica
15 LPRINT "TAVOLA PITAGORICA"
20 INPUT "Quale dimensione";N
22 REM Visualizza prima riga
25 LPRINT "      I";
30 FOR I = 2 TO N
40 LPRINT USING "   ";I;
42 LPRINT "   ";
50 NEXT I
55 LPRINT
58 REM Visualizza linea di divisione
60 FOR I=1 TO N*6+6
70 LPRINT "-";
80 NEXT I
85 LPRINT
88 REM Calcola prodotti e li visualizza, riga
per riga
90 FOR I=2 TO N
95 LPRINT USING "   ";I;
98 LPRINT "  I";
100 FOR J=2 TO N
110 LPRINT USING "   ";I*J;
115 LPRINT "   ";
120 NEXT J
130 LPRINT
140 NEXT I

```

Osserviamo che:

- la riga 40 è stata rimpiazzata dalle 40 e 42, di cui la prima specifica che il numero da visualizzare deve occupare 3 posizioni di stampa (allineato sulla destra);
- analogamente è stato operato sulle linee 95 (sostituita dalla nuova 95 e dalla 98) e 110 (sostituita dalla nuova 110 e dalla 115).

Eseguendo il programma, con dimensione 12 della tavola otteniamo su display:

```

run
Quale dimensione? 12

```

e la stampa:

TAVOLA PITAGORICA

	I	2	3	4	5	6	7	8	9	10	11	12
2	I	4	6	8	10	12	14	16	18	20	22	24
3	I	6	9	12	15	18	21	24	27	30	33	36
4	I	8	12	16	20	24	28	32	36	40	44	48
5	I	10	15	20	25	30	35	40	45	50	55	60
6	I	12	18	24	30	36	42	48	54	60	66	72
7	I	14	21	28	35	42	49	56	63	70	77	84
8	I	16	24	32	40	48	56	64	72	80	88	96
9	I	18	27	36	45	54	63	72	81	90	99	108
10	I	20	30	40	50	60	70	80	90	100	110	120
11	I	22	33	44	55	66	77	88	99	110	121	132
12	I	24	36	48	60	72	84	96	108	120	132	144

Le possibilità dell'istruzione non si fermano qui. Supponiamo di voler costruire un programma che ponga in forma tabulare dati con caratteristiche diverse, e quindi non tutti definibili con il comando di formato "ffff".

A titolo di esempio, pensiamo a un semplice programma che stampi informazioni sui pianeti del sistema solare, come il seguente, che tabula, per ogni pianeta il nome, il diametro del pianeta espresso in km, la massa del pianeta usando quella della terra come unità di misura, la distanza del pianeta dal sole, in km, il numero di satelliti.

Cominciamo a costruire la struttura più esterna del programma:

```

10 REM Stampa tabella
20 READ A$,B,C,D,E
30 REM While ci sono dati
40 IF A$="FINE" THEN 200
45 '
50 ' Qui inseriremo la stampa
55 '
190 READ A$,B,C,D,E
195 GOTO 30
200 REM Repeat
210 END
300 REM Dati relativi ai pianeti
310 REM Nome, diametro, massa rispetto alla
terra,
315 REM distanza dal sole, numero satelliti
320 DATA MERCURIO,4847,.056,58000000,0
330 DATA VENERE,12249,.817,108000000,0
340 DATA TERRA,12742,1,149675000,1
350 DATA MARTE,6748,.108,228000000,2
360 DATA GIOVE,142880,318,778000000,13
370 DATA SATURNO,120960,95,1428000000,9
380 DATA URANO,47170,14.6,2870000000,5
390 DATA NETTUNO,44990,17.26,449400000,2
400 DATA PLUTONE,5800,.1,5900000000,0
410 DATA FINE,0,0,0,0

```

Il programma effettua un ciclo WHILE leggendo ogni volta i dati relativi a un pianeta e stampandoli (la fase di stampa non è ancora presente) fino a un nome uguale a "FINE". Inseriamo ora la stampa di una intestazione prima dell'inizio del ciclo:

```

22 REM Intestazione
24 LPRINT "PIANETA      DIAMETRO  MASSA RISP.  DIS
TANZA  N.SATELLITI"
26 LPRINT TAB(12);"IN KM";TAB(23);"TERRA";
TAB(34);"DA SOLE"
28 LPRINT

```

e la stampa vera e propria, all'interno del ciclo, eliminando le istruzioni da 45 a 55:

```

64 LPRINT A$;TAB(11);B;TAB(21);C;TAB(34);D;
TAB(44);E

```

Notiamo l'uso della funzione TAB, già visto per allineare l'inizio della posizione di stampa con le posizioni dei vari "titoli" stampati con le istruzioni da 22 a 28. L'uso di TAB ci potrebbe far sperare di ottenere buoni allineamenti, invece:

PIANETA	DIAMETRO IN KM	MASSA RISP. TERRA	DISTANZA DA SOLE	N.SATELLITI
MERCURIO	4847	.056	58000000	0
VENERE	12249	.817	108000000	0
TERRA	12742	1	149675000	1
MARTE	6748	.108	228000000	2
GIOVE	142880	318	778000000	13
SATURNO	120960	95	1428000000	9
URANO	47170	14.6	2870000000	5
NETTUNO	44990	17.26	449400000	2
PLUTONE	5800	.1	5900000000	0

Le cose che "disturbano" sono:

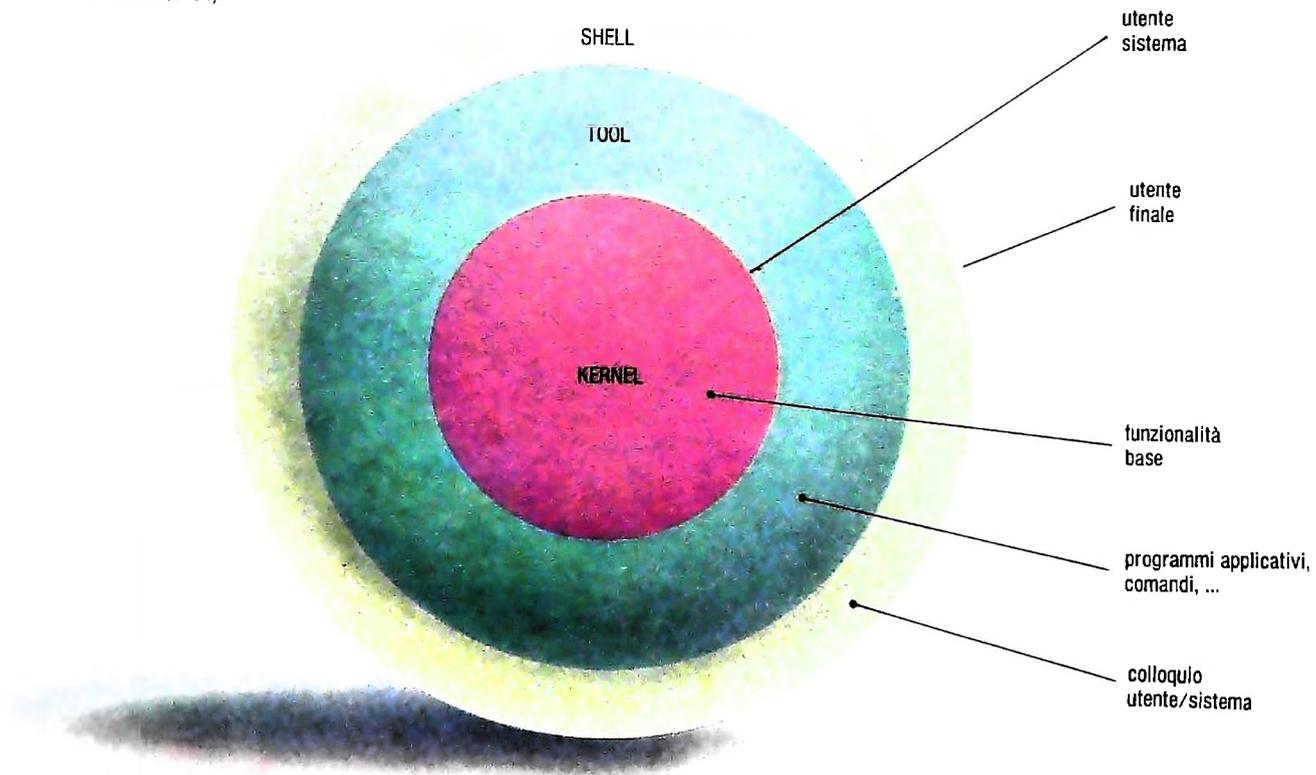
- i valori sono allineati a sinistra anziché a destra;
- i valori con decimali non sono allineati rispetto alla posizione del punto;
- i valori molto grandi sono di difficile lettura perché hanno troppi zeri: forse varrebbe la pena di ottenerli in forma esponenziale (cioè nella forma 0.58E8, che è come dire  $9.58 \cdot 10^8$ ), oltre che, come al solito, ben allineati;
- l'ultima colonna risulta completamente "sbandierata", perché in molti casi la posizione di stampa 44 (a cui avrebbe dovuto iniziare la colonna secondo la funzione TAB) era già stata invasa dal valore precedente; basterebbe cambiare il valore di TAB, ma ciò evidenzia comunque anche la scomodità di tutti questi conti.

### Cosa abbiamo imparato

In questa lezione abbiamo visto:

- l'uso dell'opzione USING nelle istruzioni PRINT e LPRINT;
- il comando di formato "££££" per la stampa di valori numerici interi.

UNIX è rappresentato qui con una struttura a "cipolla", i cui strati concentrici rappresentano le componenti del sistema, a partire dal nucleo centrale (kernel) sino allo strato più esterno (shell).



delle aziende produttrici di hardware e software-house stanno acquisendo le necessarie competenze per affrontare con successo le future richieste del mercato.

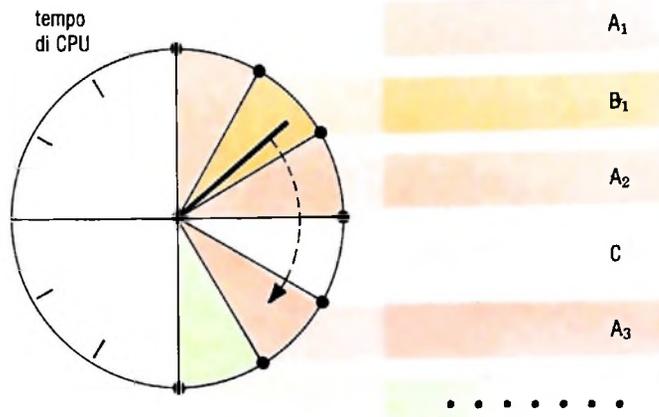
### Com'è composto Unix

Prendiamo ora in esame le componenti nelle quali è suddivisibile Unix. Il sistema è rappresentabile come una "cipolla" i cui diversi strati concentrici rappresentano le componenti (figura in alto). Questi sono tre. Il più interno è detto *kernel* (nucleo), ed è costituito da quell'insieme di elementi base del sistema che forniscono le funzionalità essenziali. È lo strato più basso del sistema operativo, quello più vicino all'hardware della macchina. L'insieme delle funzioni, dette primitive, fornite dal kernel, realizza la cosiddetta "macchina virtuale", ovvero uno pseudo-computer, al quale può accedere un programmatore di sistema (quindi non l'utente finale), per la realizzazione, per esempio, di strumenti applicativi. Il secondo strato è costituito dall'insieme dei programmi applicativi, a volte detti, in inglese, *tool* (strumenti), che realizzano i comandi del sistema, utilizzando le primitive offerte dal kernel. Lo strato più esterno è la *shell*. Questa è, in realtà, un programma come quelli presenti nell'area dei tool, ma che svol-

ge un compito molto importante: far colloquiare l'utente con il sistema, nel modo più semplice e facile possibile. È grazie alla shell che è possibile utilizzare i tool dello strato inferiore: la shell interpreterà i comandi che di volta in volta si vorranno eseguire lanciando i programmi che li realizzano. Di shell ne esistono più di una, che si differenziano per un diverso tipo di colloquio con l'utente. Così si può impiegare la shell che più si adatta alle proprie esigenze.

### Le caratteristiche principali

Volendo caratterizzare Unix rispetto all'universo dei sistemi operativi, si deve sottolineare che esso è *multi-utente* (o *multi-user*), cioè che più utenti possono contemporaneamente lavorare con il sistema dai diversi terminali collegati al calcolatore; è *multi-attività* (o *multi-processing*), cioè ogni singolo utente può far eseguire nello stesso momento più attività, ovvero programmi; inoltre Unix è a *partizione di tempo* (o *time-sharing*), cioè il sistema riserva equamente per ogni programma in esecuzione una frazione del tempo di CPU, quindi del tempo di elaborazione. Questa operazione è fatta ciclicamente fino a quando ogni programma ha termine. Questa caratteristica è tale che le diverse persone che dai propri terminali



Nell'illustrazione di questa pagina sono mostrate le caratteristiche di base del sistema operativo UNIX, che è time-sharing, multiuser, multiprocess (vedi testo). UNIX è un sistema interattivo. Nella pagina a fianco, in basso: il meccanismo delle "pipeline".

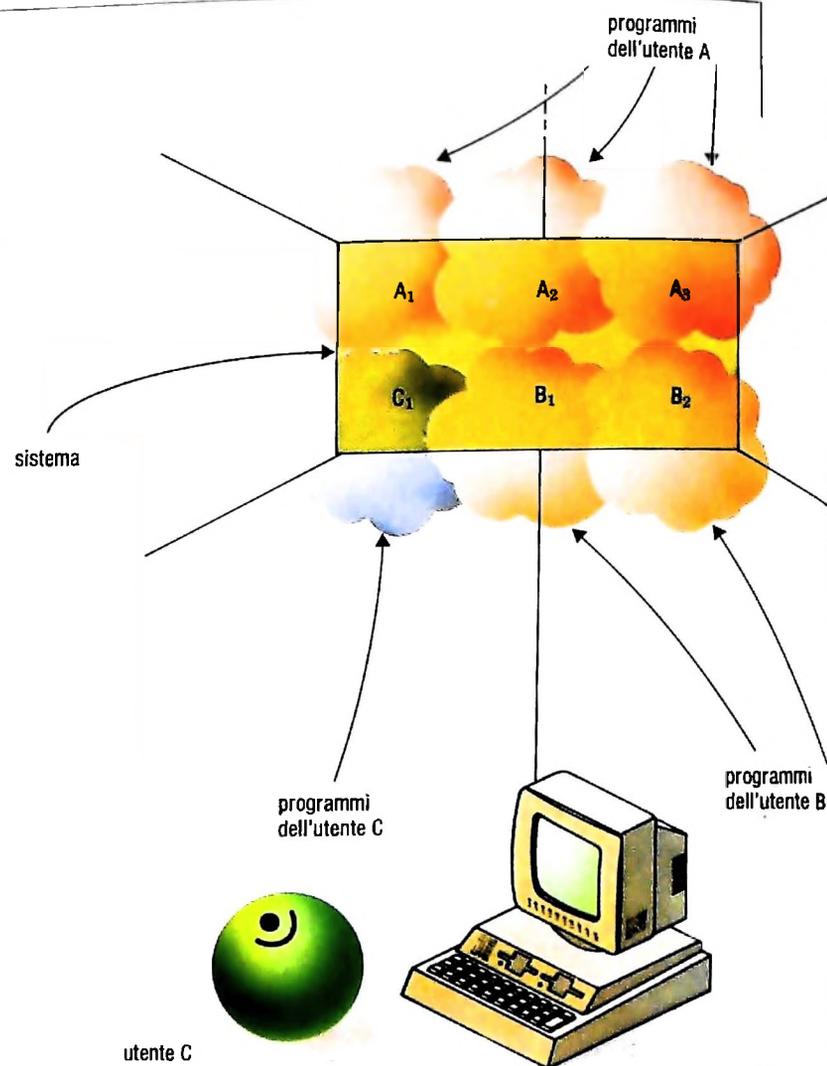
eseguono i loro programmi hanno l'impressione di avere a loro completa disposizione l'intero calcolatore, benché le risorse di questo siano, in realtà, frammentate dal sistema operativo. Unix, infatti, è un sistema tipicamente *interattivo*: l'utente "parla" con il sistema, richiedendogli lo svolgimento di alcuni compiti, e questi gli "risponde" soddisfacendo le sue richieste (figura in alto).

Comunque Unix *non* è un sistema per applicazioni in *tempo reale* (o real-time), cioè non è in grado di gestire il trasferimento di dati in tempo reale.

D'altra parte, per la ricchezza di programmi di ausilio alla programmazione presenti, e per le caratteristiche già descritte, Unix si presenta particolarmente adatto alle attività di sviluppo del software.

### Gli elementi di base

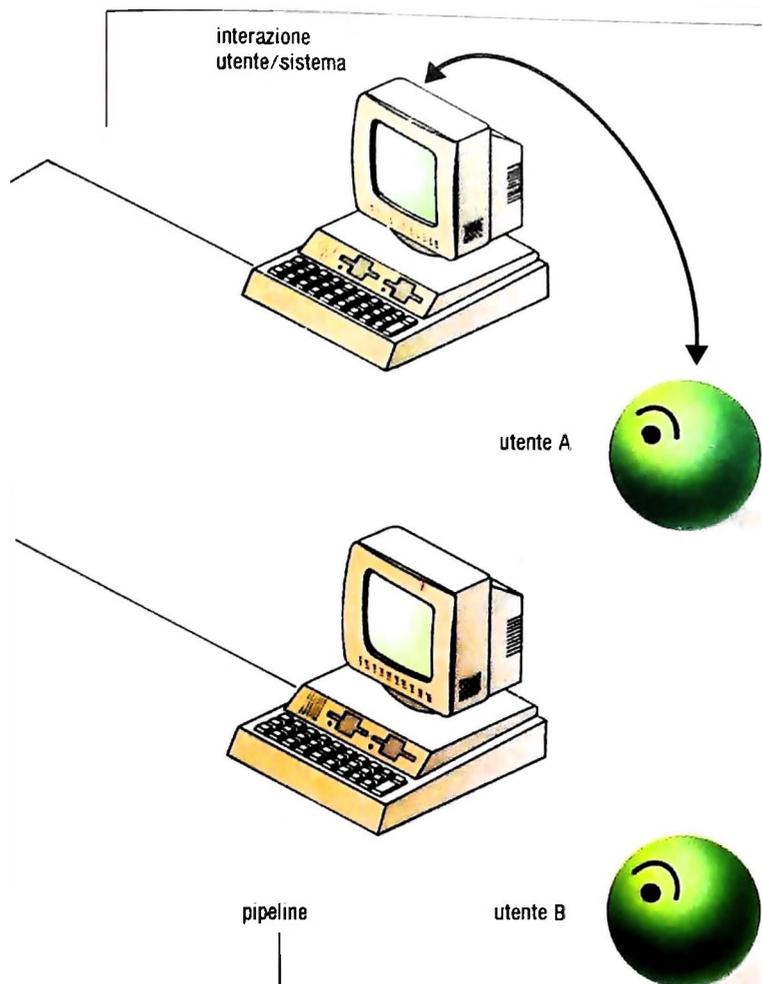
L'utente, come si è detto, è in grado di lavorare (per esempio eseguire i comandi del sistema), tramite la shell. La shell è l'interprete dei comandi: essa filtra le richieste che le giungono dal terminale, e le soddisfa dando luogo all'esecuzione dei programmi richiesti. A ogni utente è associata una specifica



shell, e da questa vengono attivati i comandi richiesti. L'insieme delle informazioni relative all'esecuzione di un comando o di un programma, oltre che l'attivazione stessa, è detto *processo*. Quindi, a ogni utente, è associato un processo relativo alla shell ed eventualmente uno o più processi che saranno relativi a quei comandi o programmi che l'utente avrà posto in esecuzione.

Un elemento centrale di Unix è il *file-system*, cioè la struttura mediante la quale è possibile memorizzare stabilmente le informazioni, dati o programmi che siano, su unità a dischi magnetici o altri supporti analoghi. In altre parole, il file-system è l'insieme dei file che tutti gli utenti del sistema archiviano, oltre a quelli già presenti nel sistema stesso. La caratteristica peculiare del file-system di Unix è di essere *gerarchico*: esso ha una struttura ad albero, nella quale i nodi, chiamati *direttrici* (in inglese "directory") contengono l'elenco dei file a essi "appesi", mentre le foglie terminali sono i file dati veri e propri.

I comandi di Unix, eseguibili tramite la shell, non sono altro che i nomi dei file che contengono i programmi archiviati nel sistema. Quindi, idealmente, l'insieme dei comandi Unix è costituito da tutti i programmi realizzati sotto Unix e archiviati nel suo file system.

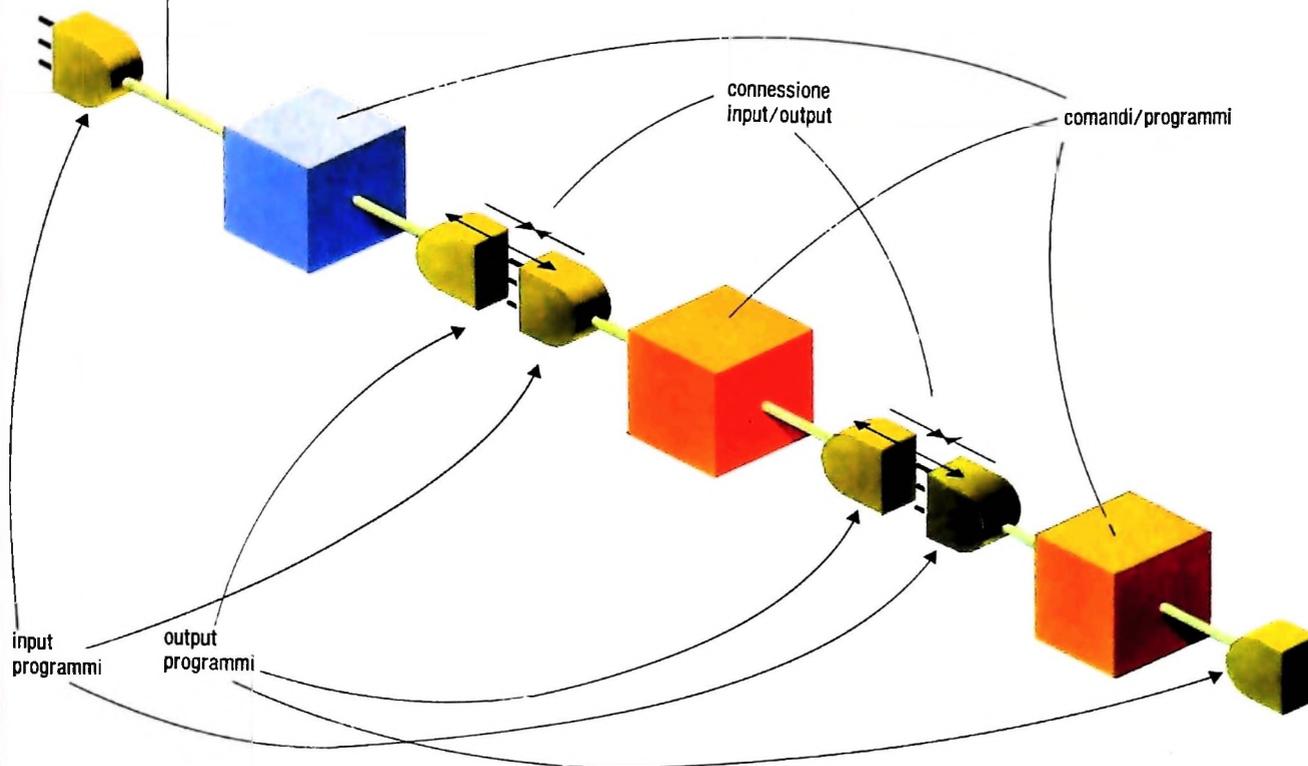


## Filosofia generale del sistema

Alcuni aspetti generali hanno permeato la realizzazione di questo sistema operativo. Innanzitutto, Unix si pone l'obiettivo di rendere facile, semplice ed elegante l'uso del sistema. Infatti, da ogni comando di Unix ci si deve attendere lo svolgimento di una sola funzione, semplice ma realizzata in modo eccellente: più comandi semplici potranno essere poi connessi tra loro, così da comporre funzioni più complesse. Questo è possibile perché la shell mette a disposizione dei meccanismi (detti *redirezioni*) mediante i quali è possibile "dirottare" l'output di un comando su un file per "salvarne" il contenuto, o direttamente nell'input di un altro comando (in questo caso si utilizzano le cosiddette *pipeline*) per poterne effettuare una elaborazione ulteriore. Non solo. Se sono necessari nuovi comandi non si deve esitare a crearli: la libreria dei tool può essere arricchita a piacimento per soddisfare meglio le proprie esigenze. Caratteristiche fondamentali di UNIX sono, quindi, la modularità e la estendibilità del sistema.

## Su quali calcolatori è possibile trovare Unix?

Fornire un elenco di tutti i calcolatori sui quali è stato reso disponibile Unix non è operazione breve. Vediamolo in sintesi. Nato sui calcolatori PDP, Unix si è poi diffuso (grazie all'apporto determinante dell'Università di Berkeley) sui calcolatori della serie VAX della Digital Equipment Co., sui calcolatori IBM Serie 1, Univac 1100, Interdata e Univac



V77. Accanto a questi calcolatori sono stati progettati calcolatori micro e super-micro con l'intento di sfruttare a fondo le caratteristiche di Unix stesso: ecco la nascita di calcolatori basati sulle famiglie di microprocessori Zilog Z8000, Motorola M68000, National 16000 ecc., quali ONYX, UX 16/20 e Samson (SGS), Momentum, Tower (NCR) e così via. Le novità più recenti del mercato riguardano le versioni Unix-like che hanno permesso a Unix di entrare anche nel settore dei personal computer: in questo ambito si inserisco-

no le iniziative dell'IBM, Tandy ed Apple. Ultimamente oltre alle tre grandi case costruttrici di microprocessori (Motorola, National Semiconductor e Intel), anche l'Honeywell e la Texas Instrument stanno promuovendo, su questa strada, iniziative concrete, mentre la Digital ha rilanciato la sua presenza nel campo Unix con il MicroVAX. Da non dimenticare, per tornare ai 'padri' di Unix, il calcolatore Bell 38 realizzato dalla A.T.&T. che supporta Unix System V, l'ultima, recentissima versione di casa Bell.

## Glossario

**Intelligenza artificiale** - lo studio e la realizzazione di programmi che consentano a un calcolatore di svolgere compiti che, se svolti da un essere umano, diremmo richiedano l'uso di intelligenza. Il confine tra ciò che richiede e ciò che non richiede intelligenza è molto sfumato, ma le ricerche sull'intelligenza artificiale possono essere viste anche come un tentativo scientifico di appurare effettivamente il significato del termine "intelligenza". I campi di studio tipici sono la produzione del linguaggio naturale, la visione e il riconoscimento di forme, i giochi (in particolare quelli che richiedono strategia).

**Interfaccia** - un dispositivo interposto fra due apparecchiature, due macchine o due dispositivi diversi, per renderne possibile la comunicazione, adeguando i segnali dell'uno a quelli dell'altro. Il termine è usato anche in senso lato e metaforico.

**LOGO** - un linguaggio di programmazione di alto livello, formulato da Seymour Papert e collaboratori presso il laboratorio di intelligenza artificiale del Massachusetts Institute of Technology. Derivato dal LISP, è un linguaggio fortemente orientato alla manipolazione simbolica e alla grafica, particolarmente adatto in ambito educativo (Papert è stato influenzato dalle teorie psicologiche e pedagogiche di Jean Piaget), tanto da essersi guadagnato fama di un linguaggio per i bambini. È, tuttavia, un linguaggio ricco e potente.

**Modem** - acronimo di MODulatore/DEModulatore. Indica un'apparecchiatura per la conversione di segnali digitali in forma analogica, adatta per la trasmissione su una linea telefonica e, nell'altro senso, per la conversione di segnali analogici in arrivo su una linea telefonica in forma digitale, adatta per l'ingresso a un calcolatore. Un modem in senso stretto effettua la conversione direttamente in segnali elettrici; un accoppiatore acustico opera invece una conversione in segnali acustici, che vengono poi trasferiti alla linea attraverso il microtelefono come i normali segnali vocali.

**NAND** - uno dei circuiti logici fondamentali di un sistema di elaborazione: equivale a un NOT AND, e quindi dà in uscita valore vero (1) se almeno uno dei due valori in ingresso è falso (0).

**NOR** - uno dei circuiti logici fondamentali di un sistema di elaborazione: equivale a un NOT OR, che dà quindi in uscita valore vero (1) se e solo se ambedue gli ingressi sono falsi (0).

**NOT** - un circuito fondamentale di un sistema di elaborazione e un operatore booleano fondamentale. NOT in inglese significa "non". Il circuito accetta in ingresso un segnale logico (0 o 1) e dà in uscita quello opposto: realizza così il comportamento dell'operatore logico, che dà il vero (1) se applicato a un operando falso (0) e viceversa.

**OR** - uno dei circuiti fondamentali di un sistema di elaborazione e uno degli operatori booleani fondamentali. OR in inglese significa "o" e il circuito e l'operatore si comportano come la congiunzione:

danno come risultato il vero (1) se almeno uno dei due segnali (uno dei due operandi) è vero (1); in caso contrario danno come risultato il falso (0).

**Ottale, sistema** - sistema di numerazione posizionale a base 8, che usa le cifre 0-7 del sistema decimale. In un numero di più cifre, ogni posizione rappresenta una diversa potenza di 8, come nel sistema decimale ogni posizione rappresenta una potenza di 10.

**Programma applicativo** - un programma che svolge una funzione direttamente pertinente alle attività dell'utente: un programma di word processing, per esempio, è un programma applicativo, come un programma per il calcolo dell'IVA o per la gestione del proprio conto bancario. Il termine è in opposizione a programma (o software) di sistema: un programma utile o essenziale per il funzionamento del calcolatore, ma non inteso direttamente allo svolgimento di un'attività per l'utente. Software di sistema è un sistema operativo, un interprete, un compilatore.

**Registro** - un piccolo gruppo di celle di memoria all'interno dell'unità centrale di elaborazione, per conservare informazioni a cui l'unità di elaborazione deve poter accedere molto velocemente. Qualunque unità centrale di elaborazione possiede numerosi registri, alcuni dei quali hanno compiti prestabiliti (il registro delle istruzioni, che conserva l'istruzione che deve essere eseguita, il contatore di programma, che conserva l'indirizzo dell'istruzione da eseguire ecc.), mentre altri sono a disposizione del programmatore, che può disporre in funzione delle necessità dei singoli programmi.

**Sistema esperto** - un sistema di calcolo programmato in modo da comportarsi come farebbe un esperto, se interrogato su problemi pertinenti al suo settore di competenza. Come un esperto, un sistema di questo tipo possiede una base di conoscenze, regole deduttive rigorose e regole euristiche non rigorose e imprecise, di tipo "intuitivo" ma corroborate dall'esperienza, e fornisce risposte alternative possibili, quantificando la probabilità relativa di ciascuna. Così, per esempio, un sistema esperto per la geologia, davanti ai dati raccolti relativamente a una data zona, potrebbe dire che ci sono x probabilità per cento di presenza di un certo tipo di giacimento, y probabilità per cento di presenza di un altro tipo di giacimento, e via dicendo.

**Struttura di controllo** - nei linguaggi di programmazione, i costrutti che definiscono l'andamento del flusso di operazioni della macchina. La struttura più semplice, e normalmente implicita, è la sequenza: le istruzioni di un programma vengono eseguite una dopo l'altra, nell'ordine in cui sono state scritte. Le altre due strutture di controllo fondamentali permettono di modificare questo flusso sequenziale: l'iterazione permette di ripetere ciclicamente un gruppo di istruzioni (per un numero prefissato di volte, oppure finché vale, o fino a che non si verifica una determinata condizione) e l'alternativa permette di eseguire l'una o l'altra fra due istruzioni (o fra due gruppi di istruzioni) a seconda che valga o meno una data condizione.

# PROBLEMI DI ANIMAZIONE AL COMPUTER

Un programma che permette di ottenere l'animazione di oggetti anche su piccoli elaboratori.



ACM SIGGRAPH

Creare una sequenza animata con l'aiuto dell'elaboratore non è un'attività semplice: richiede infatti un'accuratissima definizione della sceneggiatura, e la disponibilità di tecniche, strumenti e metodi in grado di sfruttare al massimo l'azione del computer.

Per semplificare al massimo l'analisi, vale la pena di ricordare che il computer può aiutare a elaborare immagini di tre tipi: immagini ottenute mediante la riproduzione di scene reali (con tecniche di acquisizione digitale mediante, per esempio, telecamera); immagini interamente create al computer (immagini di sintesi); infine immagini miste, in cui l'intervento di sintesi avviene per esempio su una base di immagini acquisite con tecniche digitali.

Indipendentemente dalla tecnica utilizzata per creare l'immagine singola di una sequenza vale la pena di compiere un'ulteriore classificazione imposta non tanto da particolari problematiche dettate dai metodi tradizionali di animazione, quanto dal modo con cui l'attività di animazione può venire

aiutata dal computer.

L'intento è quello di mostrare da una parte le possibilità operative, dall'altra le esigenze imposte dall'uso del computer. Infatti, in tale caso si tratta non solo di disporre della macchina (hardware) adeguata, ma anche dell'intelligenza (software) incorporata nella macchina che permette l'esecuzione delle operazioni desiderate.

## Animazione 2d

Ci si riferisce, con questa sigla, all'animazione bidimensionale. Essa consiste nella creazione di sequenze in cui l'effetto di profondità o è totalmente assente oppure è ottenuto con tecniche che simulano la prospettiva. Si differenzia dall'animazione tridimensionale (che analizzeremo più avanti) in quanto in questo secondo caso il computer dispone di tutte le informazioni necessarie a creare visualizzazioni di un mondo

```

1 REM***Esempio di animazione*****
5 REM***parametri di default*****
6 RVS=CHR$(27)+CHR$(112)'reverse video
7 NVS=CHR$(27)+CHR$(113)'normal video
10 ST=4 *Passo di incremento
20 D=10 ***numero di linee animate
30 K=2 *valore da aggiungere a RND(ST)
31 L=238:H=62
35 GOSUB 300
38 ON KEY GOSUB .....999.KEY ON
40 FORX=1TO4:Z=RND(ST):NEXT
45 XI=RND(1)*ST:XJ=RND(1)*ST
50 YI=RND(1)*ST:YJ=RND(1)*ST
60 X=RND(1)*L
70 X1=RND(1)*L
80 Y=RND(1)*H
90 Y1=RND(1)*H
100 DIM A(D,4)
110 I=1:E=2
120 CLS
130 LINE(0,0)-(L,H),1,B
150 LINE(A(E,1),A(E,2))-(A(E,3),A(E,4)),0:LINE(X,Y)-(X1,Y1)
155 LINE(L-A(E,1),A(E,2))-(L-A(E,3),A(E,4)),0:LINE(L-X,Y)-(L-X1,Y1)
160 A(I,1)=X:A(I,2)=Y:A(I,3)=X1:A(I,4)=Y1
170 I=I+1:IF I>D THEN I=1
180 E=E+1:IF E>D THEN E=1
190 IF X+XI<1 OR X+XI>L THEN XI=-1*SGN(XI)*((RND(1)*ST)+K)
200 IF X1+XJ<1 OR X1+XJ>L THEN XJ=-1*SGN(XJ)*((RND(1)*ST)+K)
210 IF Y+YI<1 OR Y+YI>H THEN YI=-1*SGN(YI)*((RND(1)*ST)+K)
220 IF Y1+YJ<1 OR Y1+YJ>H THEN YJ=-1*SGN(YJ)*((RND(1)*ST)+K)
230 X=X+XI:Y=Y+YI:X1=X1+XJ:Y1=Y1+YJ
240 GOTO150
300 REM***settaggio parametri generali*
310 CLS
320 PRINT@10,RV$+"Esempio di animazione"+NVS
330 PRINT@80,"Parametri N>ormali,P>ersonali,Q>uit ?"
340 RPS="":RPS=INKEY$.IFRPS="" THEN 340
350 IF RPS="p"ORRPS="P" THEN 370
360 IF RPS="n"OR RPS="N" THEN CLS:RETURN
365 IF RPS="q"OR RPS="Q"THEN 999ELSE BEEP:GOTO 340
370 CLS
380 PRINT@10,RV$+"Variazione parametri"+NVS
390 PRINT@80,"N. linee (>1) ":INPUT D
400 IF D<2 THEN BEEP:GOTO370ELSE D=D+1
410 PRINT@120,"Passo di incremento (>1) ":INPUT ST
420 IF ST <2 THEN BEEP:GOTO370
430 PRINT@160,"Seme di generazione (>1) ":INPUT K
440 IF K<2 THEN BEEP:GOTO370
450 PRINT@240,RV$+"Confermi S>i,N>o "+NVS
460 RPS="":RPS=INKEY$ IF RPS=""THEN 460
470 IF RPS="s" OR RPS="S" THEN CLS:RETURN
480 IF RPS="n"OR RPS="N"THEN GOTO 370 ELSE BEEP:GOTO 460
999 REM***uscita da animazione*****
1000 GOSUB 300
1010 RETURN
9999 REM***fine programma*****
10000 KEY OFF.MENU

```

tridimensionale.

L'animazione 2d è quella del tradizionale cartone animato, in cui le sequenze presentano per esempio effetti di deformazione o metamorfosi di oggetti o personaggi, oppure personaggi in movimento in scene a volte costruite su più piani fotografici.

L'intervento possibile e auspicabile del computer in questo contesto è quello di aiutare l'animatore a costruire tutte le immagini che compongono la sequenza in modo da ridurre l'enorme mole di lavoro necessario.

Un aiuto veramente utile, che comporta anche un enorme risparmio economico, può derivare dalla possibilità di colorare una serie omogenea di disegni al tratto, di variare la gamma cromatica e di visionare nuovamente la sequenza prima di effettuare la ripresa finale.

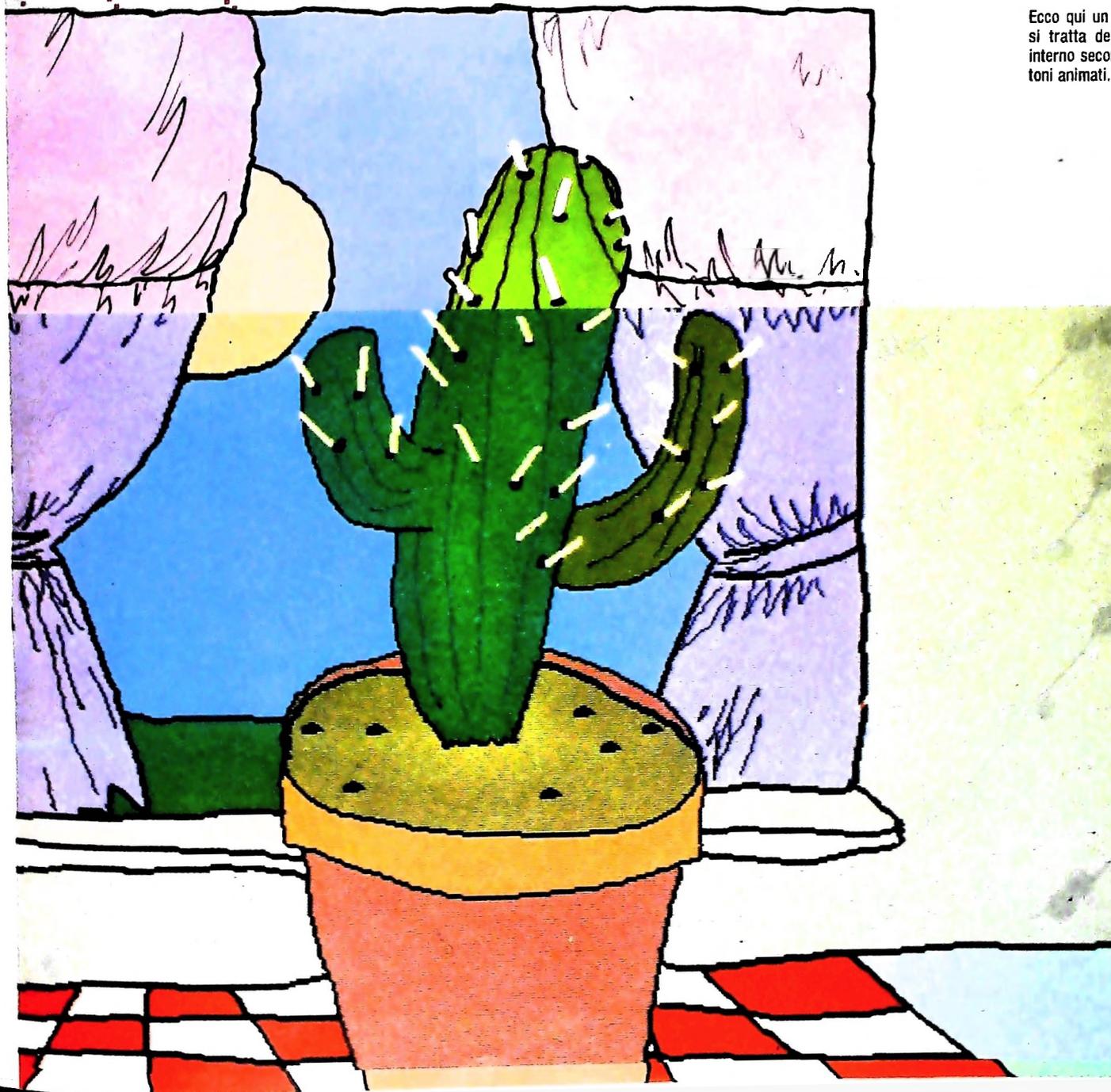
Il primo aiuto consiste nell'effettuazione del cosiddetto "inbetweening" mentre gli altri sono tradizionali aiuti resi disponibili dalle tecniche di elaborazione di immagini.

## Inbetweening

La problematica dell'inbetweening può essere sinteticamente riassunta nella possibilità di affidare all'elaboratore la generazione di configurazioni estreme. Questo è un metodo usato anche nelle tecniche di animazione manuale in cui il creativo predispone i cosiddetti fotogrammi chiave (key frame) e squadre di disegnatori eseguono tutti i disegni intermedi, necessari a ottenere movimenti il più possibile dolci e naturali.

Una soluzione automatica può apparire a prima vista semplice, per esempio adottando una regola di interpolazione tra i due fotogrammi chiave estremi. Sorgono tuttavia almeno due classi di problemi: una è dovuta al modificarsi delle quantità di informazioni presenti nella sequenza, l'altra è dovuta al tipo di interpolazione necessaria a realizzare la sequenza desiderata.

Per il primo problema si tratta di risolvere le enormi difficoltà illustrate dal seguente esempio: si supponga che la sequen-



Ecco qui un tipico esempio di painting: si tratta della rappresentazione di un interno secondo lo stile grafico dei cartoni animati.

za consista nella rotazione del viso di un personaggio: durante questa sequenza appaiono e scompaiono particolari, quali i capelli, le orecchie, gli occhi. Questi stessi particolari, se si vuole un poco di realismo, devono modificarsi in quanto durante le sequenze cambiano l'angolo e la prospettiva sotto cui vengono visti. Occorre pertanto disporre di tutte le informazioni relative al personaggio, informazioni che l'elaboratore non può inventare, ma che potrebbe proficuamente manipolare qualora il personaggio fosse descritto nella sua interezza, ovvero nella sua tridimensionalità.

Per risolvere questo tipo di difficoltà sono state proposte svariate soluzioni, tra le quali vale la pena di citare lo studio della scomposizione della sequenza in fotogrammi chiave in modo da evitare l'assenza di fotogrammi quando occorre introdurre nuove informazioni o toglierne di esistenti.

Per quanto riguarda il secondo problema si pensi, a titolo di esempio, al movimento che compie un braccio quando fa un cenno di saluto. In tale caso l'avambraccio ruota avanti e indietro attorno a un punto (il gomito), e la mano descrive un arco di circonferenza. Ricostruire le posizioni intermedie in questo caso richiede l'uso non di interpolatori lineari (che genererebbero figure intermedie tutte distribuite lungo una linea retta), ma piuttosto interpolatori adeguati a ricostruire percorsi circolari.

## Painting

La problematica della colorazione di disegni al tratto può essere facilmente risolta mediante le cosiddette tecniche di "painting".

Queste tecniche consistono in un primo luogo nella possibilità di colorare con la tinta desiderata (che nei sistemi più sofisticati può essere scelta in una gamma di 16 milioni di colori) una qualsiasi regione del disegno. Spesso i sistemi di painting permettono di simulare l'intera gamma di azioni disponibili a un grafico, quali il tecnigrafo, il set di penne, matite, pennarelli, che permettono di realizzare un vasto insieme di segni e colori.

Con le tecniche di painting è dunque possibile creare un disegno che soddisfa due importanti requisiti: è nella memoria del computer, e quindi può essere facilmente elaborato, e inoltre, è in una forma che può, con relativa facilità, essere diffusa nel circuito televisivo.

## Il programma

Il programma permette di ottenere un'animazione in tempo reale: si propone infatti di fornire il movimento a un semplice oggetto costituito da un gruppo di linee il cui numero può anche essere definito dall'utente.

L'effetto che si ottiene è il movimento di un numero variabile di linee lungo lo schermo: quando uno dei due punti della prima linea tocca un lato del video esso funge da fulcro intorno al quale ruotano tutte le altre fino a ottenere un cambiamento di direzione di tutto l'oggetto.

Nella stesura del listato si è fatto pesantemente uso della funzione RND(x) che, generando traiettorie casuali, aumenta notevolmente la suggestività dell'animazione.

## Il listato

Linee 1-31: poiché non è necessario definire i parametri dell'oggetto, ma è possibile usare dei valori di default, si prevede in queste linee di assegnare i valori delle costanti. Per una maggiore flessibilità del programma si sono assegnati anche i valori delle dimensioni della finestra di visualizzazione ( $x = 238, y = 62$ ).

Linee 35 e relativa routine: per permettere l'introduzione e la variazione dei parametri di default si è prevista la chiamata e la gestione della routine di cui alle linee 300-480.

Linee 300-480: in questa procedura si entra per selezione iniziale oppure mediante la pressione del tasto funzione F8. Si rende possibile la variazione di alcuni parametri che influiscono direttamente sulla visualizzazione:

- passo incremento: rappresenta un valore usato per la generazione della distanza fra le linee dell'oggetto che vogliamo animare;

- seme generazione: è questo un parametro che permette di aumentare la suggestività degli effetti ottenuti. La sua variazione influisce direttamente sul generatore di numeri casuali;

- numero linee: aumentando o diminuendo questo valore si varia il numero di linee visualizzate; si faccia però attenzione alla variazione della velocità di esecuzione.

Linea 38: poiché lo schermo deve essere mantenuto libero il più possibile per la visualizzazione si è prevista l'uscita dal programma con la pressione del tasto funzione F8 (la relativa routine di gestione è alle linee 999-1010). A tale scopo si deve abilitare la lettura dei tasti funzionali con il comando KEY ON.

Linee 40-120: i valori di partenza e i successivi incrementi vengono fissati in maniera casuale (con la limitazione imposta dai parametri costanti).

Linee 130-160: in queste linee si ottiene il tracciamento dell'oggetto nella posizione corrente che è individuata dai valori iniziali oppure dai valori generati con una tecnica semi-casuale nelle linee successive.

Linee 170-240: è questa la parte più importante di tutto il programma. In queste linee si procede ai controlli sugli estremi delle linee al fine di individuare il contatto di un estremo con i bordi del video; se ciò avviene si varia la direzione della linea imponendone una rotazione.

Il presente listato è da considerarsi come un piccolo esempio di animazione in tempo reale. Si noti che il listato è stato costruito per ottenere un effetto visivo che illustri la possibilità di animare oggetti anche su piccoli elaboratori.

Ogni ulteriore espansione del programma esula dal presente corso di basic poiché in questo caso sarebbe infatti utile una traduzione del programma in codice macchina, al fine di ottenere migliori risultati in esecuzione.

— UN NUOVO MODO DI USARE LA BANCA. —

# Conto corrente più

## TANTI PENSIERI IN MENO CON IL CONTO CORRENTE "PIU" DEL BANCO DI ROMA.

Essere cliente del Banco di Roma vuol dire anche essere titolari del conto corrente "più". Un conto corrente più rapido: perché già nella maggior parte delle nostre filiali trovate gli operatori di sportello che vi evitano le doppie file.

Più comodo, perché potete delegare a noi tutti i vostri pagamenti ricorrenti: dai mutui all'affitto, dalle utenze alle imposte.

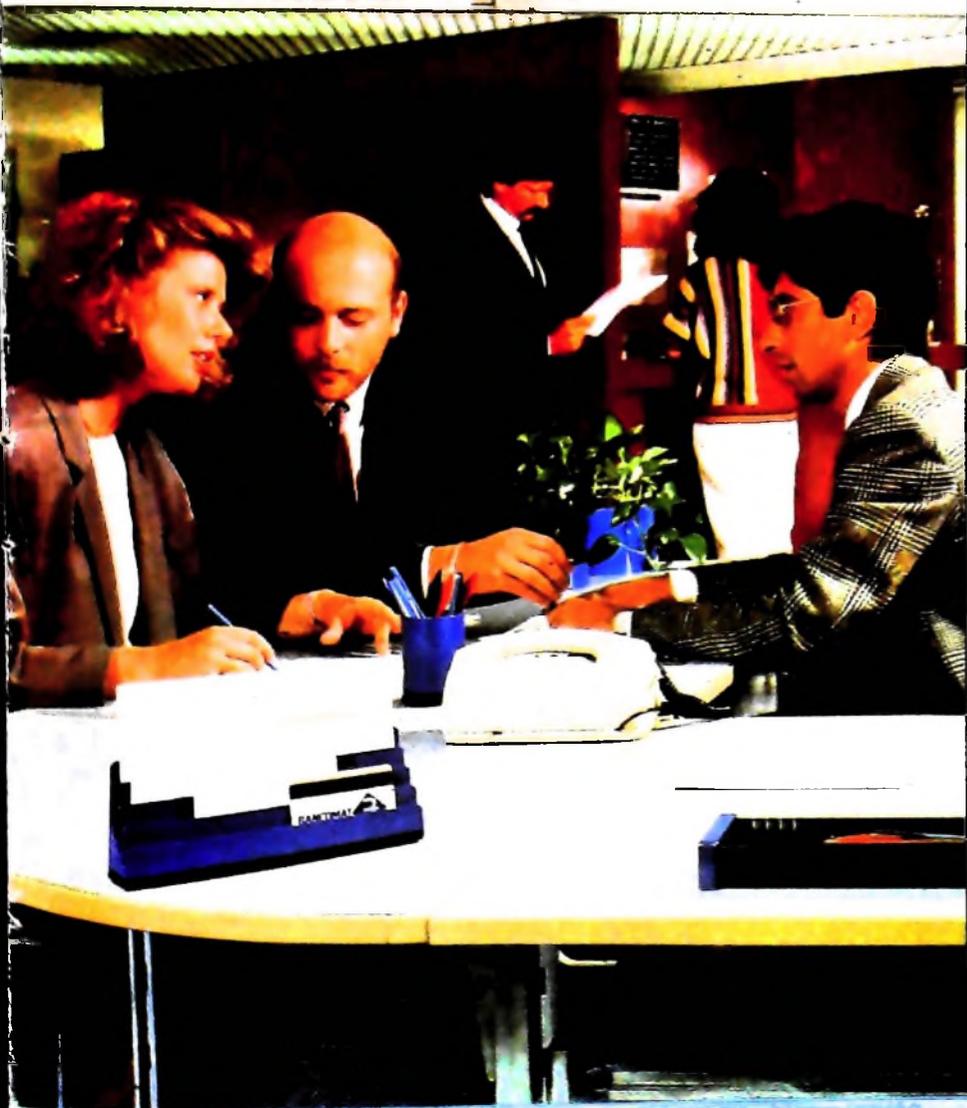
Più pratico, perché consente l'utilizzo del sistema di prelievo automatico Bancomat e l'ottenimento della carta di credito.

Inoltre un servizio utilissimo, soprattutto per imprenditori e commercianti denominato "esito incassi", consente di avere comunicazione dell'eventuale insolvenza entro solo cinque giorni dalla scadenza. Una opportunità veramente speciale.

Più sicuro, perché con una minima spesa potrete assicurarvi contro furti e scippi mentre vi recate in banca o ne uscite.

Veniteci a trovare, ci conosceremo meglio.

 **BANCO DI ROMA**  
CONOSCIAMOCI MEGLIO.



Olivetti M10 vuol dire disporre del proprio ufficio in una ventiquattrore. Perché M10 non solo produce, elabora, stampa e memorizza dati, testi e disegni, ma è anche capace di comunicare via telefono per spedire e ricevere informazioni. In grado di funzionare a batteria oppure collegato all'impianto elettrico, M10 mette ovunque a disposizione la sua potenza di memoria, il suo display orientabile a cristalli liquidi capace anche di elaborazioni grafiche, la sua tastiera professionale arricchita da 16 tasti funzione.



Ma M10 può utilizzare piccole periferiche portatili che ne ampliano ancora le capacità, come il micro-plotter per scrivere e disegnare a 4 colori, o il registratore a cassette per registrare dati e testi, o il lettore di codici a barre. E in ufficio può essere collegato con macchine per scrivere elettroniche, con computer, con stampanti. Qualunque professione sia la vostra, M10 è in grado, dovunque vi troviate, di offrirvi delle capacità di soluzione che sono davvero molto grandi. M10: il più piccolo di una grande famiglia di personal.

## PERSONAL COMPUTER OLIVETTI M10

# L'UFFICIO DA VIAGGIO



Anche in leasing con Olivetti Leasing.

**olivetti**

Per informazioni rivolgersi ai negozi contrassegnati da "Olivetti M10 Punto di Vendita" o al viaggiatore coupon a Olivetti, Divisione Personal Computer, Via Meravigli 12, 20123 Milano.

NOME/COGNOME \_\_\_\_\_  
 VIA \_\_\_\_\_  
 CAP/CITTA' \_\_\_\_\_  
 TELEFONO \_\_\_\_\_