

Spediz. in abbonamento postale GR II/70 L. 2.200
(...)

58 CORSO PRATICO COL COMPUTER

422105

F4

F5

F6

F7

diretto da GIANNI DEGLI ANTONI

è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**



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 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
Eidos (TIZIANO BRUGNETTI, BRUNO MOTTA, FRANCESCO SALVI),
CLAUDIO PARMELLI, 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

Segretarie di Redazione
RENATA FRIGOLI
LUCIA MONTANARI

Corso Pratico col Computer - Copyright © sul fascicolo 1985 Gruppo Editoriale Fabbri, Bompiani, Sorzogno, Etas S.p.A., Milano - Copyright © sull'opera 1984 Gruppo Editoriale Fabbri, Bompiani, Sorzogno, 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. 58 - 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.

CORSO PRATICO COL COMPUTER

Volume Quinto

CORSO PRATTICCO

Diretto da GIANNI DEGLI ANTONI

Q

Z

R

Y

A

S

F

G

H

C

B



COL COMPUTER

FABBRI EDITORI

U 4

I 5

O 6

P

=
,
_

J 1

L 3

ENTER

%
ù

?
,
0

:
[

8

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

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 BREGGÉ

Segretarie di Redazione

RENATA FRIGOLI

LUCIA MONTANARI

Corso Pratico col Computer

Copyright © 1984 Gruppo Editoriale Fabbri, Bompiani,
Sonzogno, Etas S.p.A., Milano

Prima Edizione 1984

LA POSTA ELETTRONICA (I)

Come recuperare la funzionalità dell'informazione con riduzione dei costi oltre che dei tempi di lavoro.

Dopo aver illustrato i principi e le tecniche della trasmissione dati, ne presentiamo ora alcune applicazioni. Cominciamo dalla Posta Elettronica.

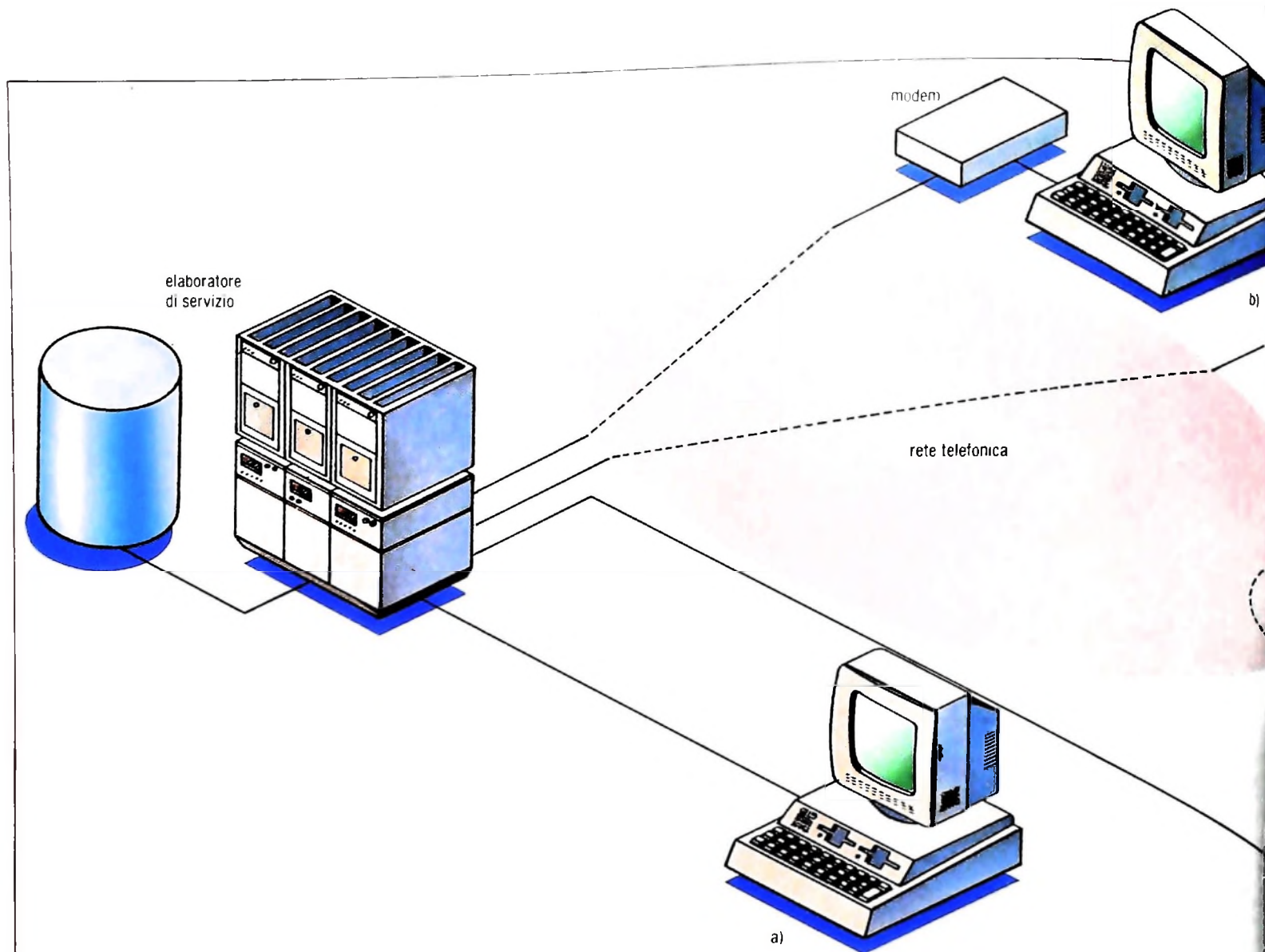
La strumentazione occorrente è formata da un terminale o elaboratore, per esempio M10; da un software di comunicazione asincrona che permette di "adeguarsi" al protocollo di trasmissione utilizzato, in questo caso TELCOM, che si trova già inserito nel computer; da un modem o accoppiatore acustico, e infine da una normale linea telefonica, con apparecchio telefonico annesso.

Come in un normale servizio postale anche nella Posta Elettronica esiste un postino (la linea telefonica), un ufficio po-

stale e una casella, in questo caso elettronica (rappresentati dal sistema o computer centrale che governa il servizio) che permettono l'inoltro, l'immagazzinamento e la ricerca di messaggi di qualsiasi natura, persino di listati di programma. Quest'ultimo caso, con M10 è reso possibile memorizzando il listato in modalità document, quindi con suffisso .DO. Lo stesso verrà trattato come un qualsiasi testo ASCII e quindi trasmesso, ricevuto e fatto eseguire senza doverlo ridigitare. La funzionalità del sistema di posta elettronica è data dall'assoluta mancanza di ritardi nell'inoltro della corrispondenza, se si eccettuano quelli dovuti alla procedura di collegamento e alla trasmissione, e dalla possibilità di tenere i



INFOMARK è un valido supporto alle funzioni tradizionali della posta elettronica. Caratteristica principale di questo sistema è la capacità di organizzare un'archiviazione centrale in cui l'utilizzatore può creare propri archivi con garanzie di riservatezza, e gestirne le informazioni da ogni suo terminale attraverso la rete Data-mont, in Italia o all'estero.



messaggi archiviati e ricercabili, secondo criteri specifici, nella propria casella elettronica (o mail-box) senza preoccupazioni di smarrimenti o refresh.

Di seguito analizzeremo il sistema della DATAMONT, l'INFOMARK (INFORMATION MARKET) che con concentratori installati in tutti i capoluoghi e maggiori città italiane, per un totale di 33 unità, permette di effettuare la comunicazione evitando la costosa teleselezione. Infatti un utente di Palermo che desidera inviare un messaggio all'utente di Milano, potrà utilizzare il concentratore di Palermo, mentre per il destinatario sarà sufficiente accedere a quello di Milano per ritrovare il messaggio memorizzato nella propria mail-box.

Modalità d'accesso a INFOMARK

Per accedere al sistema di Posta Elettronica è necessario possedere una casella elettronica presso il concentratore, casella che viene fornita all'utente dietro presentazione di domanda per l'utilizzo del servizio.

Dopo essere sicuri del corretto collegamento tra computer e modem, lanciare (nel caso M10) il programma TELCOM e dimensionare la stringa di comunicazione secondo il seguente elenco:

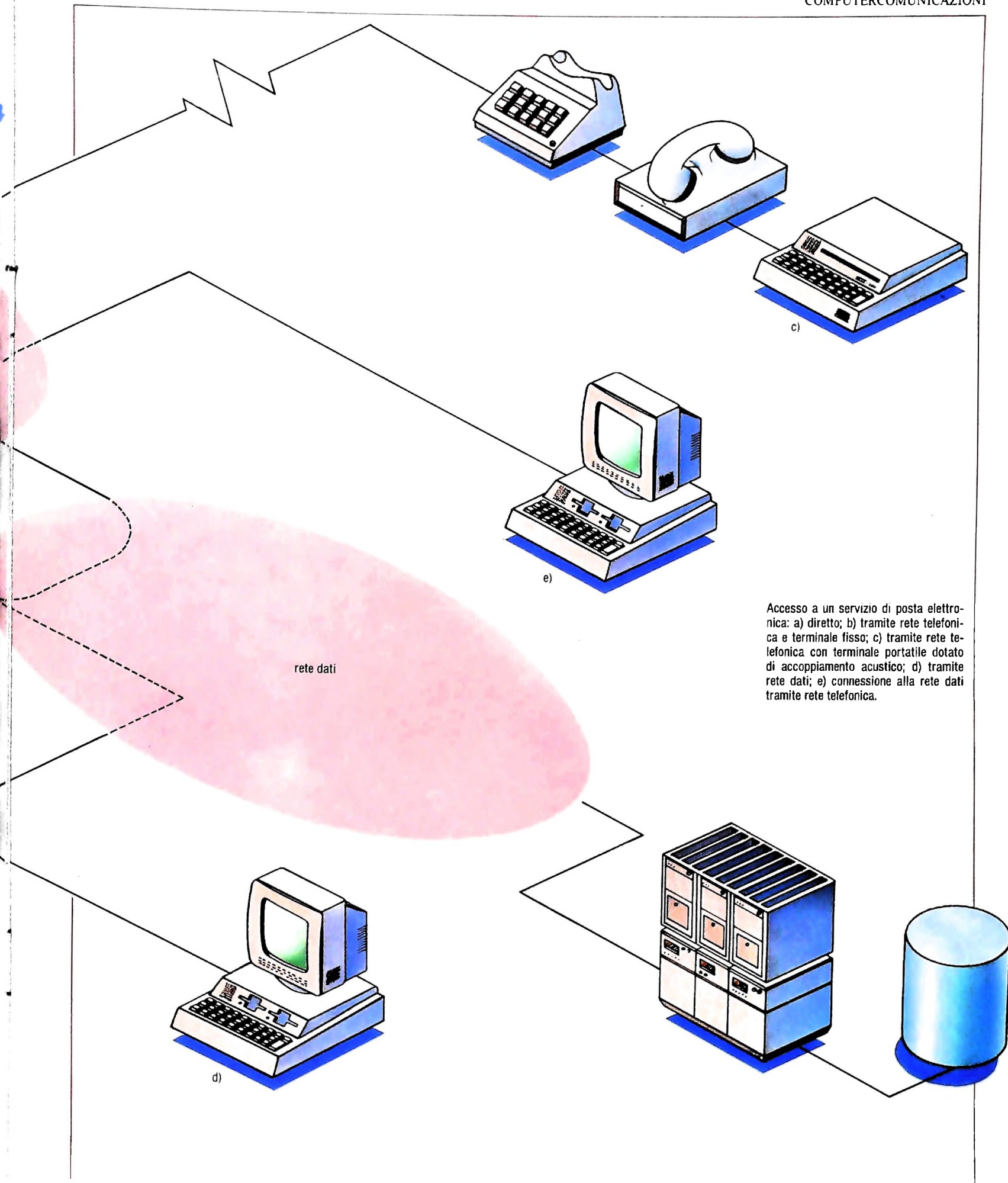
BAUD RATE	300 baud
LUNGHEZZA PAROLA	7 bit
PARITÀ	EVEN
STOP BIT	1
STATUS LINE	E (Enable)

ricordando inoltre di inserire l'opzione FULL DUPLEX e disinserire l'ECHO.

Nel caso di altri elaboratori, lanciare il programma di comunicazione asincrona corrispondente e dimensionare il Protocollo di trasmissione secondo i valori citati.

Espletata tale operazione, comporre il numero telefonico relativo al concentratore più vicino, al ricevimento della portante inserire il microtelefono (cornetta) negli appositi alloggiamenti dell'accoppiatore acustico oppure, nel caso del modem, commutare lo stesso da telefono a dati.

Nella fase immediatamente successiva al collegamento inviare un CR (ritorno carrello) e il sistema, dopo essersi definito, chiederà il nominativo dell'utente e quindi, dopo aver ricevuto corretta risposta, la password.



Accesso a un servizio di posta elettronica: a) diretto; b) tramite rete telefonica e terminale fisso; c) tramite rete telefonica con terminale portatile dotato di accoppiamento acustico; d) tramite rete dati; e) connessione alla rete dati tramite rete telefonica.

Mentre nella prima fase (quella in cui viene richiesto il nominativo dell'utente) il sistema provvederà a visualizzare (tramite ECHO) i caratteri trasmessi dall'utente, per ragioni di segretezza ne sospenderà l'effetto durante l'introduzione della password. Se la stessa risulterà non corretta, il sistema la richiederà al massimo per due volte e poi annullerà il collegamento. Una volta inviata la password corretta il sistema comunicherà quando è stato effettuato l'ultimo collegamento con il nome dell'utente e indicherà se, da allora, qualcuno ha tentato inutilmente di collegarsi con quel nome (password errate).

Queste sono informazioni che garantiscono una completa sicurezza e permettono di individuare subito eventuali situazioni anomale.

Dopo questa serie di dati, INFOMARK comunicherà quanti messaggi sono stati ricevuti dopo l'ultimo collegamento, cioè il numero di messaggi che dovranno essere letti, e infine chiederà un comando da parte dell'utente.

Tutti i comandi e le risposte che verranno date al sistema dovranno terminare sempre con un CR.

Ogni qualvolta si dovranno generare dei messaggi, il sistema, in funzione del comando impartito, richiederà, in maniera guidata, la compilazione dei soli campi indipendenti provvedendo automaticamente al riempimento dei restanti; i campi non essenziali potranno essere non riempiti, in questo caso basterà inviare direttamente il carattere di ritorno carrello.

Vediamo di seguito un esempio di connessione, tenendo conto che in maiuscolo e in neretto sarà evidenziato tutto ciò che è prodotto dal sistema, mentre indicheremo con caratteri minuscoli tutti i comandi che verranno impartiti dall'utente; il CR indica l'invio di un ritorno carrello. L'esempio inizia dal momento in cui il collegamento telefonico è stato effettuato.

HIRES: info —CR—

INFOMARK SERVICE — BL52 8 — THURSDAY,

JANUARY 28, 1985 15:39

PLEASE TYPE YOUR NAME: esempio —CR—

PASSWORD: prova —CR—

THANK YOU. YOUR LAST LOGIN WAS MONDAY,

JANUARY 16, 1985 13:00 ITL

YOU HAVE 10 NEW MESSAGES.

COMMAND:

Archivi gestiti dal sistema

Ogni utente potrà creare un numero a piacere di file propri e gestirli come meglio crede.

INFOMARK da parte sua provvederà a creare e a gestire

automaticamente solo quattro file, e questi sono:

- 1 - "UNREAD" il file dei messaggi da leggere
- 2 - "UNSENT" il file dei messaggi da trasmettere
- 3 - "DELETED" il file dei messaggi da cancellare
- 4 - "CHRONO" il file cronologico dei messaggi

1 - Nel file UNREAD il sistema colloca e gestisce tutti i messaggi che vengono trasmessi per l'utente in oggetto. Quando l'utente si collega al sistema questo gli comunicherà il numero dei messaggi giacenti in tale archivio.

A mano a mano che sarà effettuata la lettura dei messaggi, gli stessi saranno tolti automaticamente dal file e verranno inseriti, oltre che in eventuali archivi indicati dall'utente, nel file CHRONO.

2 - Tutti i messaggi creati utilizzando gli appositi comandi vengono inseriti automaticamente, oltre che in eventuali archivi specificati dall'utente, nel file UNSENT.

Quando avverrà la trasmissione dei messaggi stessi, il sistema provvederà a rimuoverli da questo archivio per inserirli nel CHRONO.

3 - Tutti i messaggi che l'utente cancella vengono temporaneamente immagazzinati nel file DELETED (fino a che la sessione in corso non viene chiusa); questo permette, in caso di errore, l'eventuale recupero di messaggi che altrimenti sarebbero definitivamente annullati.

4 - Nel file CHRONO vengono memorizzati tutti i messaggi ricevuti e letti nonché quelli trasmessi al fine di avere un archivio completo e cronologico di tutti i messaggi scambiati con gli altri utenti.

È inoltre possibile creare degli archivi a proprio uso anche se i messaggi contenuti non verranno mai trasmessi; per esempio si può creare un file AGENDA dove l'utente elenca le attività che dovrà svolgere o gli appuntamenti che dovrà rispettare.

Funzioni di controllo

Vediamo ora le principali funzioni di controllo che sono a disposizione del sistema INFOMARK.

Il sistema utilizza i tasti sia singolarmente che in molteplici combinazioni.

Si usufruisce singolarmente di un tasto (ESCAPE) per annullare l'esecuzione di un comando e del tasto (DELETE o RUBOUT) per cancellare i caratteri erroneamente battuti; si usufruisce, invece, di più tasti battuti contemporaneamente al fine di:

- visualizzare la linea su cui si è operata la cancellazione del (o dei) caratteri. Naturalmente verrà esclusa la visualizzazione dei caratteri cancellati e le barre che li contengono (CONTROL R);
- fermare la visualizzazione dell'output (scroll) per esaminarlo con calma (CONTROL S);
- riprendere la visualizzazione dell'output (CONTROL Q);
- fermare lo scroll senza possibilità di riprenderla nel caso in cui non interessi più (CONTROL O);
- annullare la riga che si sta digitando (CONTROL Z).

Lezione 57

Ancora sulla gestione degli errori

Nelle lezioni precedenti abbiamo visto come sia possibile prevenire la segnalazione di errori da parte del BASIC, attraverso l'analisi carattere per carattere di dati d'ingresso e attraverso il controllo attento di valori accettabili. Abbiamo anche visto come sia possibile distinguere chiaramente i momenti in cui gli errori vengono definiti, rilevati e trattati, rendendo il programma più ordinato. Infine, abbiamo imparato il meccanismo di dichiarazione di occorrenza di errore con l'istruzione ERROR. Proseguiamo con l'esame del trattamento degli errori.

Supponiamo di costruire un programma come il seguente:

```
Ok
100 I=1
110 S=0
120 S=S+I
130 NEXT I
```

Si tratta di un programma grossolanamente sbagliato, ove a una parte iniziale che sembra voler gestire un ciclo iterativo basato sull'indice I, per totalizzare in S i vari valori di I, segue una parte finale che sembra voler gestire l'indice I con un NEXT.

Dalla sua esecuzione ci aspettiamo errori. E se la segnalazione fosse:

```
110 S=0
120 S=S+I
130 NEXT I
run
Pollo! Non vedi che hai messo un NEXT senza
il FOR precedente, ALLA LINEA 130
Ok
```

penseremmo forse che qualcuno ci sta imbrogliando, in quanto, non solo il tono del messaggio è insolito, ma inoltre il messaggio è scritto in italiano (pure cosa insolita), e, per quanto cerchiamo di sfogliare un qualunque manuale di BASIC, non esiste nessuna traccia di un tale messaggio. In realtà con l'uso delle istruzioni che già abbiamo visto, ci siamo sostituiti al BASIC nelle segnalazioni d'errore.

Infatti il meccanismo dell'istruzione ERROR non è nuovo per il BASIC, ma lo stesso che il linguaggio usa per segnalare gli errori, reso disponibile all'utente.

Se un programma usa come istruzione iniziale una ON ERROR, tutti gli errori che occorreranno, siano essi dovuti a istruzioni ERROR esplicitamente dichiarate dal programmatore, siano essi dovuti a condizioni d'errore abitualmente reperite dal BASIC (come un errore di sintassi, una divisione per zero ecc.), comporteranno il rinvio al sottoprogramma di trattamento degli errori: se in questo è previsto il trattamento di errori non solo legati a istruzioni ERROR per codici d'errore definiti dal programmatore, ma anche a codici di errore già esistenti il trattamento così definito attraverso il nostro sottoprogramma si sostituirà a quello tradizionale di sistema.

Esaminiamo un tipico errore segnalato dal BASIC: il NEXT senza FOR del programma precedente produrrebbe, in condizioni normali, il messaggio:

```
?NF AT LINE 130
```

e ciò corrisponde proprio all'errore numero 1 (lo possiamo vedere dalla tabella presentata un paio di lezioni fa, ma lo rivedremo anche in questa lezione).

L'occorrenza di un errore mette a disposizione la variabile ERR che contiene il numero d'errore, e anche ERL, che contiene il numero di linea in cui l'errore è stato reperito: con queste due informazioni potremo costruire programmi in cui i messaggi d'errore del BASIC sono completamente sostituiti dai nostri, ma con la stessa capacità di individuare l'errore al numero di linea in cui esso è occorso. Supponiamo di costruire un programma in modo che inizi con le seguenti istruzioni:

```
1 DIM A$(255)
2 ON ERROR GOTO 5000
3 GOSUB 4000 'Traduzione codici errore
100 I=2
110 I=I*I
120 GOTO 110
130 NEXT I
```

L'array A\$ ha lo scopo di contenere tutti i messaggi d'errore che noi vogliamo ridefinire, la routine di trattamento d'errore deve essere posizionata alla linea 5000, e un sottoprogramma alla linea 4000 si occuperà di leggere in A\$ i vari messaggi d'errore corrispondenti a ciascun codice.

La routine di trattamento dell'errore potrà essere:

```
5000 'Trattamento errori
5005 PRINT A$(ERR);" ALLA LINEA";ERL
5010 END
```

A fronte dell'errore il cui codice numerico è indicato in ERR, viene visualizzato il messaggio d'errore che a esso è stato fatto corrispondere, e il numero di linea in cui la condizione d'errore è stata trovata.

Il sottoprogramma di lettura dei codici d'errore potrà essere allora il seguente:

```
4000 'Lettura codici errore tradotti
4010 FOR I=1 TO 255
4020 A$(I)="Errore indefinito"
4030 NEXT I
4031 READ I
4032 'While ci sono dati do
4034 IF I=0 THEN 4900
4036 READ A$(I),I
4038 GOTO 4034
4039 'Endwhile
4040 DATA 1,"NEXT senza FOR"
4041 DATA 2,"Errore di sintassi"
4042 DATA 3,"RETURN senza GOSUB"
```

```

4043 DATA 4, "Dati insufficienti"
4044 DATA 5, "Chiamata funzione errata"
4045 DATA 6, "Valore troppo grande"
4046 DATA 7, "Memoria insufficiente"
4047 DATA 8, "Linea non definita"
4048 DATA 9, "Indice fuori dai limiti"
4049 DATA 10, "Array ri-dimensionato"
4050 DATA 11, "Divisione per zero"
4051 DATA 12, "Comando immediato errato"
4052 DATA 13, "Variabile di tipo errato"
4053 DATA 14, "Memoria per stringhe esaurita"
4054 DATA 15, "Stringa troppo lunga"
4055 DATA 16, "Espressione stringa troppo
complessa"
4056 DATA 17, "Impossibile proseguire"
4058 DATA 19, "Manca istruzione RESUME"
4059 DATA 20, "RESUME senza routine d'errore"
4061 DATA 22, "Operando mancante"
4062 DATA 23, "Errore di input/output"
4089 DATA 50, "Errore interno"
4090 DATA 51, "Numero di file errato"
4091 DATA 52, "File non trovato"
4092 DATA 53, "File già aperto"
4094 DATA 54, "Lettura oltre fine file"
4095 DATA 55, "Nome file errato"
4096 DATA 56, "Istruzione immediata in un file"
4097 DATA 57, "Troppi files"
4098 DATA 58, "File non aperto"
4099 DATA 0
4900 RETURN

```

Esso, inizialmente, pone in ogni stringa di A\$, in corrispondenza di qualunque codice d'errore, il messaggio "Errore indefinito"; successivamente legge un codice d'errore in I e il corrispondente messaggio in A\$(I); la lettura si arresta quando, nei dati, viene reperito un codice d'errore pari a zero; in tal modo tutti i codici d'errore che hanno qualche messaggio associato vanno a sovrapporre tale messaggio a quello precedentemente inserito come "Errore indefinito", che resterà solo per eventuali errori definiti dall'utente, ma senza un corrispondente messaggio.

Osserviamo che nella numerazione degli errori (e, corrispondentemente, nelle linee di BASIC), esistono alcuni "buchi":

- errori 18 e 21: il BASIC non fornisce un messaggio;
- errori da 24 a 29: la versione attuale del BASIC non fornisce alcun messaggio, ma si suppone che ulteriori estensioni del linguaggio potranno richiedere l'indicazione di nuovi errori, che useranno questi codici;
- errori da 59 a 255: i codici d'errore non sono usati dal BASIC e sono disponibili all'utente per la definizione di propri errori.

Il comportamento di un programma con un simile "contorno" operativo fornisce quindi messaggi d'errore in italiano, e con una chiarezza molto maggiore a quella a cui siamo abituati: è sufficiente che il programma vero e proprio sia inserito nel "buco" tra la linea 3 e il sottoprogramma 4000.

Vediamo alcuni esempi: il programma

```
100 PRINT "*"
```

fornisce il messaggio

```
run  
Errore di sintassi ALLA LINEA 100  
Ok
```

Il programma

```
100 PRINT 10/0
```

fornisce il messaggio

```
run  
Divisione per zero ALLA LINEA 100  
Ok
```

Il programma

```
100 I=2  
110 I=I*I  
120 GOTO 110
```

fornisce il messaggio

```
run  
Valore troppo grande ALLA LINEA 110  
Ok
```

quando, a furia di moltiplicare I per se stesso, il valore risultante va in "overflow", cioè supera la capacità di rappresentare un valore di una variabile del calcolatore. Notiamo che il sottoprogramma, avendo tutti i "buchi" per i codici definibili dall'utente, finisce per trattare in modo assolutamente uniforme errori definiti dall'utente e già impliciti nel BASIC. Ecco spiegato finalmente il fantasioso e irriverente messaggio d'errore mostrato all'inizio della lezione!

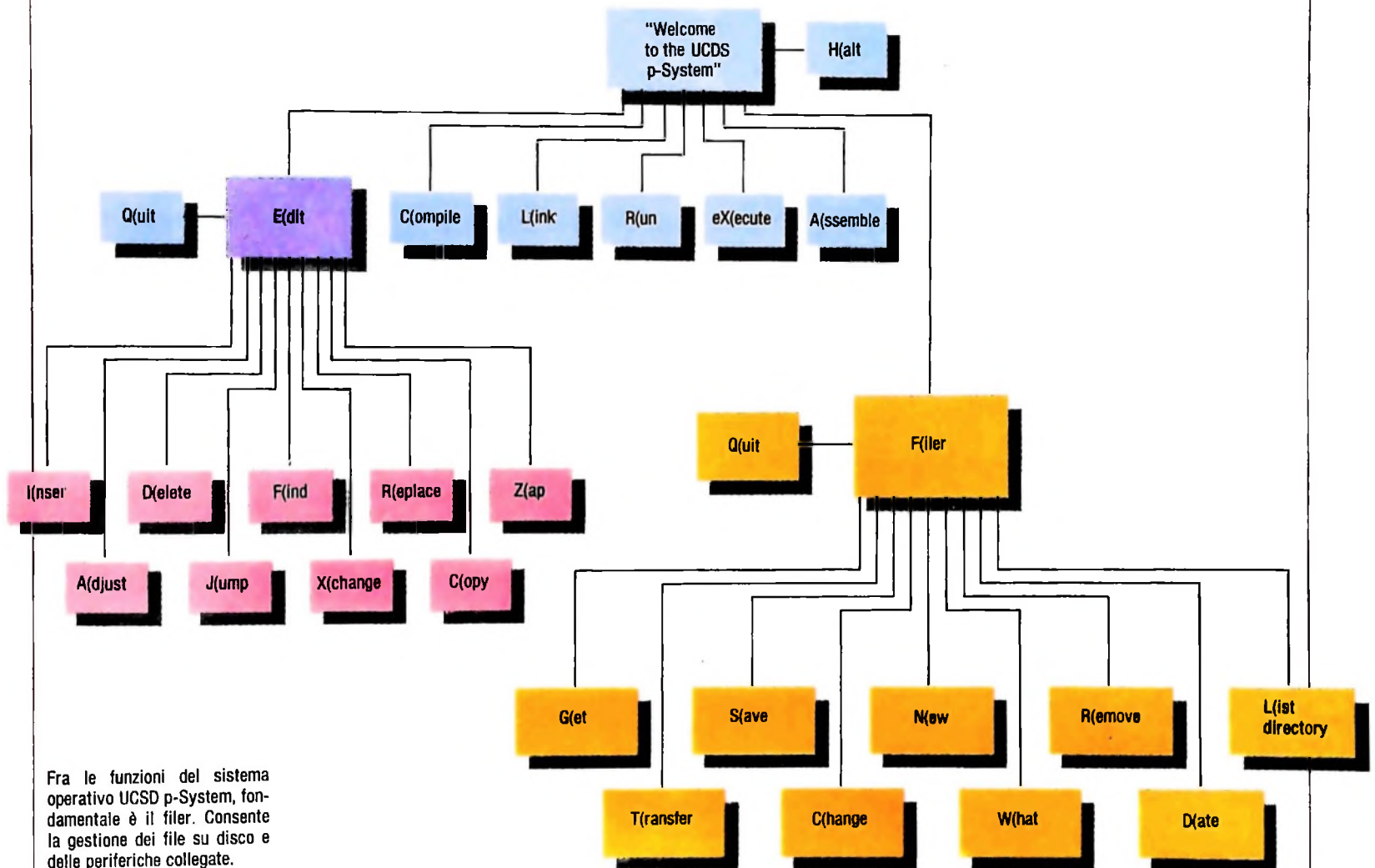
Cosa abbiamo imparato

In questa lezione abbiamo visto:

- l'uso della ON ERROR per sostituirsi ai messaggi del BASIC;
- l'uso della variabile ERL che contiene il numero di linea a cui è stato incontrato un errore, accanto alla variabile ERR che contiene il relativo codice d'errore

IL SISTEMA OPERATIVO UCSD P-SYSTEM (III)

Continuiamo l'analisi delle funzioni del sistema.



Fra le funzioni del sistema operativo UCSD p-System, fondamentale è il filer. Consente la gestione dei file su disco e delle periferiche collegate.

Il filer

Un altro fondamentale componente del p-System, anch'esso richiamato con un solo tasto (F) a partire dal menù principale, è il *filer* (figura in alto).

Il filer consente la completa gestione dei file su disco (trasferimenti, cancellazioni, cambiamenti di nome ecc.) e delle pe-

riferiche (video, stampante, e altre eventuali). Con gli stessi comandi è possibile trattare tanto i file utente, quanto gli stessi file che compongono il p-System: per esempio, è possibile copiare il compilatore Pascal e l'editor su un proprio disco da utilizzare per la preparazione di programmi. Vediamo ora il funzionamento dei comandi del filer.

Il comando G consente di copiare un file di testo in un'area

```
Remove what file ? ?.text
```

il carattere "?" indica che vanno cancellati tutti i file il cui nome termina per ".TEXT", dietro controllo dell'operatore

```
Remove TENTA.TEXT ? y
Remove LTG.TEXT ? y
Remove MORRACIN.TEXT ? y
Remove TSUTIL.TEXT ? y
Remove TEXTSTUF.TEXT ? y
Remove TSUTIL2.TEXT ? y
Remove QUESTION.TEXT ? y
Remove STAT.TEXT ? y
Remove TORNEO.TEXT ? y
Update directory ?
```

In alto l'utente chiede la rimozione dei file il cui nome termina col suffisso .TEXT. In basso: il sistema elenca tutti i file selezionati e chiede conferma per l'eliminazione.

speciale di lavoro, allo scopo di operare modifiche (mediante l'editor), eventuali compilazioni, prove ecc. Al termine delle operazioni, le modifiche possono essere riportate sul file (comando S), oppure annullate (comando N).

Il comando R consente di eliminare uno o più file da un disco. È possibile effettuare l'eliminazione di gruppi di file sulla base del loro nome. L'esempio di questa pagina mostra l'utilizzo di tale comando: l'utente batte R (Remove) seguito da "? . TEXT" (figura in alto, a sinistra).

Il carattere "?" ha un significato speciale detto, in inglese *wildcard*, cioè jolly, e sta a indicare una qualsiasi sequenza di caratteri. L'utente ha richiesto cioè di eliminare tutti i file il cui nome termina col suffisso ". TEXT", indipendentemente dalla prima parte del loro nome.

Il sistema risponde (figura in alto, a destra) elencando i nomi dei file selezionati uno a uno e chiedendo una conferma ("y" (yes) o "n" (no)). Tale conferma selettiva può essere evitata utilizzando il carattere "*" invece di "?": in questo modo l'utente può confermare o annullare soltanto l'intera operazione. Il comando L consente di visualizzare il contenuto di un disco (pagina a fianco, in alto).

Il comando T consente di trasferire uno o più file (con lo stesso meccanismo descritto a proposito del comando R) da un disco a un altro, o da un disco verso una periferica di output (per esempio la stampante).

Il comando C consente di modificare il nome di uno o più

file, o il nome associato a un disco. Altri comandi consentono poi di visualizzare e modificare la data del giorno, riordinare i file sul disco ecc.

Lo spooler

Come abbiamo accennato in precedenza, il p-System comprende un certo numero di programmi di utilità che vengono eseguiti utilizzando il comando eX (ecute), come un qualsiasi programma utente. Questi programmi effettuano funzioni anche molto importanti, ma non così frequenti da essere incluse tra i comandi del menù principale.

Tra questi programmi di utilità, esaminiamone uno: lo *spooler*. Lo spooler consente di effettuare la stampa di file durante l'uso normale del p-System (in particolare nei tempi morti in cui il sistema attende l'immissione di dati o comandi da tastiera), senza bloccare l'intera attività.

Con questo programma è possibile, per esempio, inviare un file alla stampante e utilizzare contemporaneamente l'editor per effettuare modifiche su un altro file.

Nella figura di pagina a fianco, in basso, vediamo il menù che compare quando il programma spooler viene richiamato. Il comando P(rint viene utilizzato per richiedere la stampa di un file. Esso può essere usato ripetutamente, poiché lo spooler è in grado di gestire una coda di 21 file da stampare.


```

File: Blet, S(ave, W(hat, N(ew, L(dir, R(em, C(hng, T(rans, D(ate,? [C.11]
VARIE:
SYSTEM.COMPILER 102 25-Jan-82
SYSTEM.ASSMBLER 60 9-Jan-82
8086.OPCODES 5 22-May-81
8086.ERRORS 11 22-May-81
8087.FOPS 3 22-May-81
SYSTEM.LINKER 26 19-Jul-81
TENTA.TEXT 6 22-May-84
LTG.TEXT 12 23-May-84
LTG.CODE 4 23-May-84
QUESTION.CODE 6 5-Sep-84
MORRACIN.TEXT 12 5-Jul-84
TSUTIL.TEXT 18 11-Sep-84
TEXTSTUF.TEXT 8 11-Sep-84
TEXTSTUF.CODE 6 11-Sep-84
TSUTIL.CODE 4 11-Sep-84
TSUTIL2.TEXT 8 11-Sep-84
TSUTIL2.CODE 3 11-Sep-84
QUESTION.TEXT 14 12-Sep-84
STAT.TEXT 10 12-Sep-84
TORNEO.TEXT 22 8-Oct-84
TORNEO.CODE 8 8-Oct-84
21/21 files(listed/in-dir), 358 blocks used, 282 unused, 282 in largest

```

```

Spool: P(rint, D(elete, L(list, S(uspend, R(esume, A(bort, C(lear,? [IV.1 3e]

```

```

Spool: P(rint, D(elete, L(list, S(uspend, R(esume, A(bort, C(lear,? [IV.1 3e]
TENTA.TEXT
STAT.TEXT
TORNEO.TEXT

```

In alto: usando il comando L viene visualizzato il contenuto di un disco. In basso: ecco il menù che compare sul video quando si richiama il programma spooler. Tramite lo spooler, è possibile stampare durante l'uso normale del sistema p-System, cioè senza bloccare l'intera attività.

Il comando D(elete consente di eliminare un file dalla coda di stampa.

Il comando L(ist elenca i file presenti sulla coda di stampa (si noti che l'eventuale file in fase di stampa non si trova già più nella coda).

Il comando S(uspend ferma temporaneamente la stampa, che può essere ripresa mediante il comando R(esume.

Il comando A(bort viene utilizzato per fermare la stampa ed

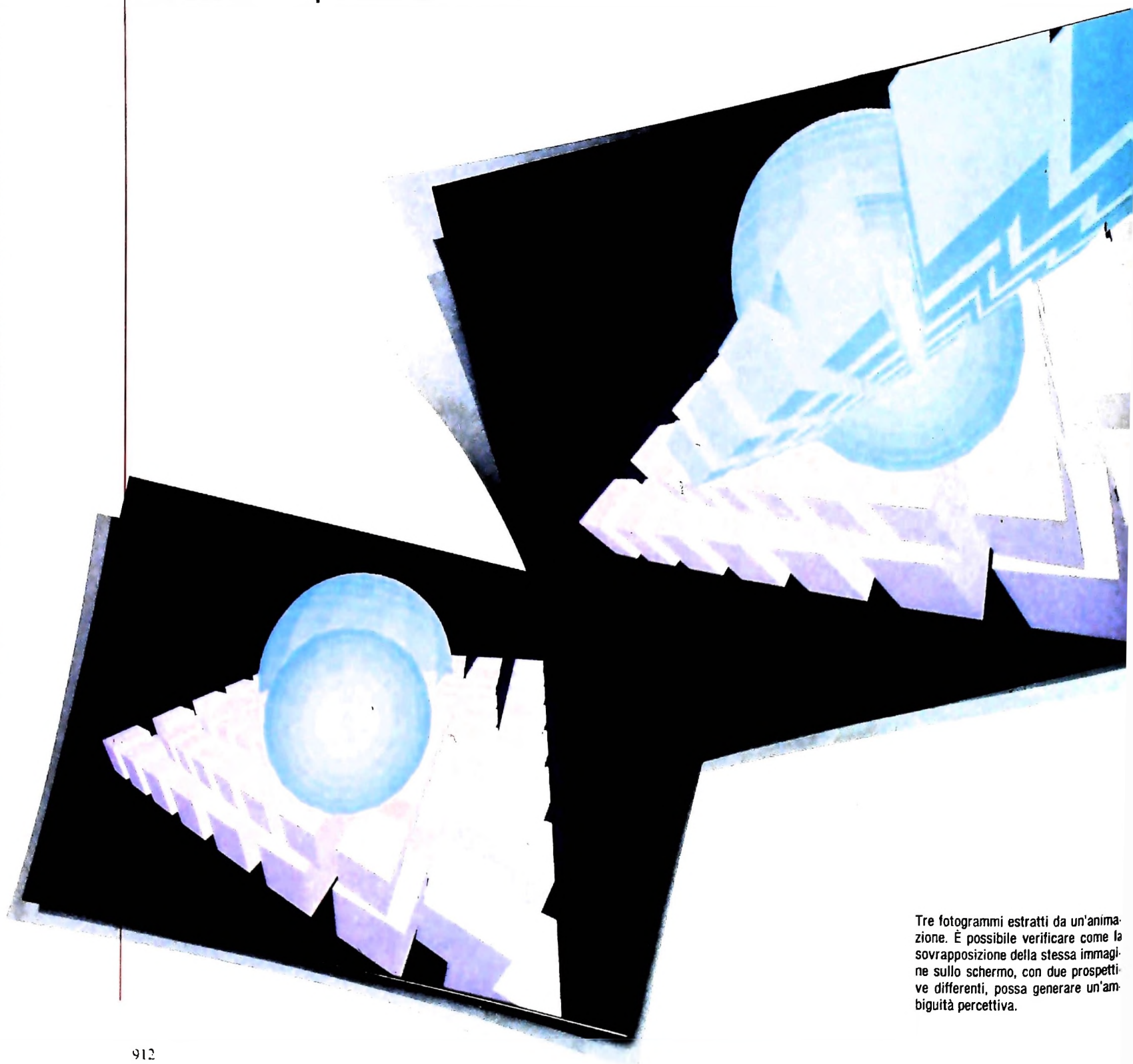
eliminare dalla coda il file in corso di stampa.

Il comando C(lear elimina tutti i file dalla coda di stampa.

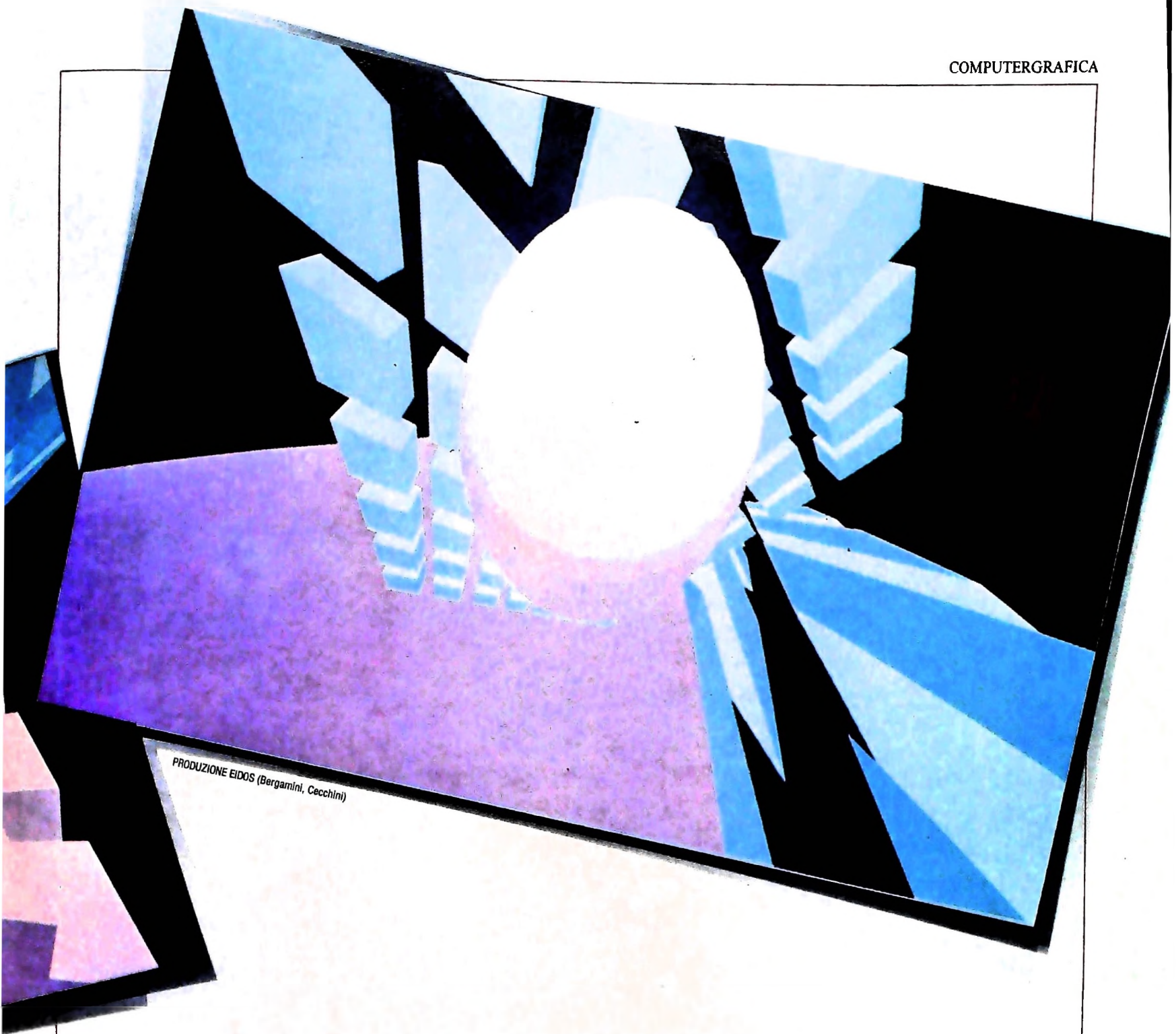
Infine, il comando Q(uit termina l'esecuzione del programma spooler. In questo momento inizia l'effettiva stampa dei file presenti nella coda, che procede fino a che la coda non è esaurita, o fino a che non si verificano errori in fase di stampa, o fino a che l'utente non richiami nuovamente il programma spooler.

VERSO L'ANIMAZIONE AL COMPUTER

'Un programma per realizzare un orologio digitale
con suoneria e pendolo animato.



Tre fotogrammi estratti da un'animazione. È possibile verificare come la sovrapposizione della stessa immagine sullo schermo, con due prospettive differenti, possa generare un'ambiguità percettiva.



PRODUZIONE EIDOS (Bergamini, Cecchini)

Una delle più spettacolari e divertenti discipline eidomatiche, la Computer Animation, sta assumendo un interesse sempre più crescente sia nel campo dell'intrattenimento che nelle applicazioni tecniche, quali la simulazione di processi naturali o artificiali.

Allo stato attuale dell'evoluzione hardware, le caratteristiche grafiche che facevano riconoscere un'immagine come prodotta dall'elaboratore (scarsa scelta di colori, generalmente i tre fondamentali, bassa definizione, scalettatura delle linee diagonali, stile "a quadretti") sono superate. Tramite gli schermi moderni, ad altissima risoluzione, con software adeguato, siamo molto vicini a immagini di verosimiglianza prossima a quella dell'immagine fotografica. È venuta così in evidenza la caratteristica che realmente contraddistingue la grafica computerizzata: le immagini prodotte possono cambiare continuamente a seconda del programma preparato o

rispondendo alle azioni dell'utente.

Questo fatto sta inducendo molti studi di grafica, di pubblicità e, in generale, coloro che producono immagini, a dotarsi di strumentazioni computerizzate.

Perché questo? Le ragioni stanno in una delle caratteristiche di fondo dell'elaboratore: quella di poter ripetere innumerevoli volte un'azione, dopo che questa è stata descritta all'inizio dal programmatore, e di poterne variare i parametri. Questa caratteristica, fondamentale nella produzione industriale, si rivela quanto mai utile anche nel campo cosiddetto "creativo". Ecco che il grafico, l'impaginatore, il disegnatore o l'inchiostatore sono liberi di modificare il loro originale in un attimo, di ottenere copie diversificate da confrontare e di eseguire tutte le prove che ritengono necessarie per ottenere il miglior risultato.

Dopo aver acquisito i mezzi per gestire l'immagine statica

l'eidomatica si è mossa per conquistare anche la dimensione temporale.

La caduta dei prezzi delle unità di calcolo, lo sviluppo di software applicabile con facilità anche al di fuori dei centri di ricerca e, soprattutto, l'interesse a creare un mondo sintetico dove si possa assistere all'accadere di fenomeni del tutto immaginari ma perfettamente coerenti con le definizioni che l'utente ha imposto fanno dell'animazione il campo più fertile dell'eidomatica.

La "Computer Animation" è sostenuta soprattutto dalle richieste delle agenzie di pubblicità che, forti del successo dello stile computerizzato, propongono sempre più spezzoni di filmati realizzati all'elaboratore. Effetti speciali e scenografie particolari sono, molte volte, più facili da realizzare e meno costosi. Con un computer è possibile non solo definire rapidamente un oggetto come un pupazzo (costruito con primitive grafiche tipo sfere, cilindri ecc. *in tre dimensioni*) e colorare un disegno immesso via tavoletta grafica, ma è possibile ripercorrere tutta la genesi di un cartone animato definendone solo le immagini principali e lasciando al programma il compito di completare il lavoro.

Per facilitarne l'uso, il software di animazione è costruito attorno alla metafora della "cinepresa artificiale", cioè presenta all'utente dei comandi che ricalcano quelli di una normale cinepresa che risultano più familiari e comprensibili.

Le trasformazioni geometriche e prospettiche dell'immagine, ottenute per vie matematiche all'interno del programma, sono specificabili dall'esterno con nomi come "zoomata avanti e indietro", "carrellata", "panoramica". Un programma completo per animazioni fornisce inoltre una serie di possibilità utili in cinematografia, come l'inserimento di scritte per comporre titoli o per i cartelli del ciak e la conservazione dei dati della ripresa: numero del fotogramma, lunghezza pellicola utilizzata, posizione della cinepresa ed elenco delle trasformazioni effettuate e da effettuare.

Vogliamo qui dare un semplice approccio al concetto di animazione, facendo uso di M10 e del linguaggio Basic a disposizione. Verrà presentata in seguito anche un'applicazione animata più versatile e più completa, dotata di un semplice interfaccia utente mediante l'utilizzo della tastiera.

Aspetti implementativi

Sfruttando le possibilità grafiche di M10 è possibile ottenere degli effetti che, anche se non si possono considerare di utilità pratica, possono aiutare a capire meglio il funzionamento dell'elaboratore che si sta analizzando.

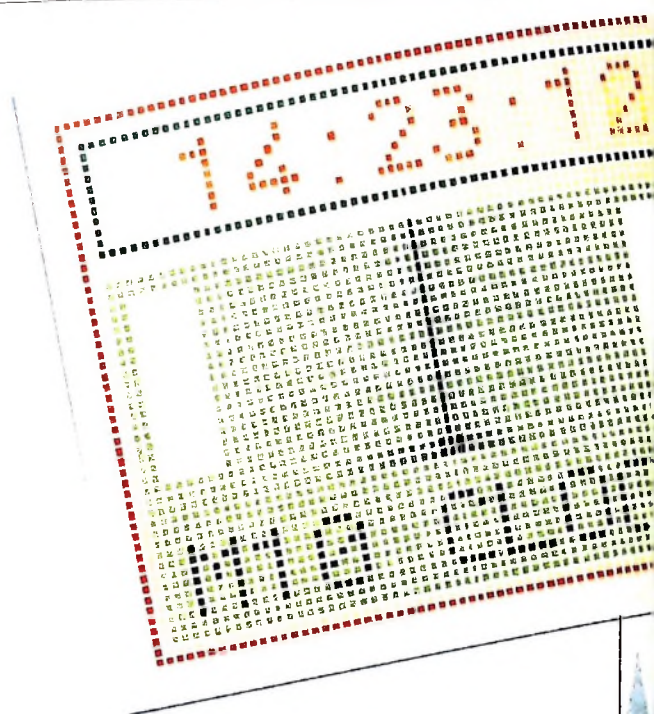
A questo punto del corso è ormai nota la maggior parte dei comandi del linguaggio Basic di M10 ed è quindi inutile dilungarsi in quelle definizioni e in quei concetti già molte volte commentati. È quindi opportuna solo una breve introduzione che permetta di spiegare la tecnica seguita nella stesura del presente programma: una applicazione che unisce la grafica all'uso del suono e che si appoggia notevolmente alla funzione `time$` prevista dal Basic M10.

All'interno di tale variabile sono contenuti i valori numerici

dell'ora con la precisione di un secondo essendo essa aggiornata periodicamente dal calcolatore che offre quindi la possibilità di avere un aggiornato orologio digitale a più funzioni. In unione alla `time$`, infatti, sono disponibili anche le variabili `date$` e `day$` rispettivamente per la data e il giorno.

Tutte queste funzioni possono essere modificate dal Basic come se si stesse trattando una comune variabile di stringa aggiornandone periodicamente i valori.

L'aspetto principale che spinge a inserire il presente programma nelle lezioni di grafica è rappresentato dal fatto che l'orologio disegnato sul video di M10 è (anche se molto stilizzato) dotato di un pendolo animato da movimento ritmico, il cui moto è controllato direttamente dalla variabile `time$` (si usa dire che "batte il secondo"). Il problema che si vuole risolvere è di ottenere l'animazione in tempo reale dell'asta del



La 'Computer Animation' consente la realizzazione "in vitro" di scenografie particolari ed effetti speciali, che risulterebbero altrimenti molto costosi in termini di tempo e di denaro. L'uso del software di animazione è in questo momento sostenuto soprattutto dalle richieste delle agenzie di pubblicità.

pendolo (nel caso di M10 si può animare un oggetto soltanto cancellandolo e disegnandolo accanto in maniera periodica).

Il programma

Passiamo ora a esaminare il listato cercando di rendere modulari le singole procedure.

Si inizia con una chiamata a una routine specifica per il disegno dell'orologio (linee 3000-3060) a cui fa seguito la scrittura di una frase di modalità "reverse video" (caratteri bianchi su sfondo nero). Tale funzione è ottenuta mediante la trasmissione di una sequenza di caratteri di controllo a M10:

Reverse video = chr\$(27) + chr\$(112)

ESC + p
Normal video = chr\$(27) + chr\$(113)
ESC + q

Il programma permette inoltre di rientrare al menù principale premendo il tasto funzione F8, questo rende indispensabile l'abilitazione dei tasti funzionali (linee 25-30).

Le costanti di programma vengono tutte definite alla linea 40 e rappresentano:

- C = nota usata per le ore
- D = durata delle note
- F = passo di decremento per animazione
- P = posizione centrale del pendolo.

Poiché il presente programma simula un orologio dotato di sonoro (la nota melodia che viene ripetuta con incremento di

un quarto d'ora dal Big Ben) è necessario esaminare quale delle quattro fasi è necessario suonare e quindi la quantità di note da produrre (linee 45-120).

La parte che ci interessa maggiormente è alle linee 125-200 nelle quali si esamina la variazione della parte terminale della stringa TIME\$ (i secondi).

Si decide di proseguire l'animazione del pendolo in un senso fino a quando la variazione di una cifra dei secondi non provoca l'inversione del movimento rovesciando la direzione del moto dell'asta (linea 150).

Alla linea 150 si provvede inoltre a emettere un suono che indica lo scatto del secondo.

Le linee 170-180 producono l'animazione vera e propria poiché si preoccupano di disegnare il pendolo cancellandolo dalla posizione immediatamente vicina e simulando così il movimento.

A cadenza di un secondo viene inoltre aggiornato il video (stampa di time\$ alla linea 190).

Il resto del listato si preoccupa della gestione del suono:

990-1040 = legge e suona le note nel numero indicato in

```

2 REM**Orologio digitale con suoneria
3 REM***e pendolo animato per M10****
10 CLS:GOSUB 3000
15 REM*** stampa scritta reverse*****
20 PRINT @256,CHR$(27) CHR$(112);"M10
CLOCK";CHR$(27);CHR$(113);
25 REM***abilita i tasti funzione*****
30 TIME$ ON:ON KEY GOSUB .....500:KEY
ON:PRINT @315,"FINE";
35 REM***settggio parametri costanti***
40 C=4697:D=20:F=-1:P=122
45 REM***definisce i quarti per suono**
50 H=VAL(MID$(TIME$,4,5))
60 IF H<15 THEN M=1
70 IF H=>15 THEN M=2
80 IF H=>30 THEN M=3
90 IF H=>45 THEN M=4:H=1:
ELSE H=0
100 H=H+VAL(LEFT$(TIME$,2))
110 IF H>23 THEN H=0
120 GOSUB 1100
125 REM***stampa ora ed animaz. pendolo
130 H$=TIME$
140 IF H$=TIME$ THEN 140
150 IF H$<>TIME$ THEN H$=TIME$:F=F/(-1):
SOUND 9500 1
160 P=P+F:IF P<117 OR P>127
THEN P=P-F
170 LINE (P-F,20)-(P-F,41),1:LINE (P-F-1,42)-
(P-F+1,44),1,BF
180 LINE (P,20)-(P,41),0:LINE (P-1,42)-(P+1,44),
0,BF
190 PRINT @57,TIME$;
200 GOTO 150

```

dipendenza dai quarti;

1045-1080 = suona le ore;

1090-1130 = analizza la funzione time\$ rilevandone la variazione delle cifre;

2000 = contiene sotto forma di DATA le note della melodia.

La linea 500 è incaricata della terminazione del programma che ha fine mediante la pressione del tasto funzione F8 che provoca un rientro al menù principale.

Conclusioni

Il programma presentato riassume molte delle funzioni già affrontate e offre una possibilità di collegare fra loro tutte quelle che rendono M10 una macchina davvero versatile.

Si ribadisce ancora la mancanza di un effettivo scopo applicativo anche se è opportuno notare che con la presente realizzazione si è introdotto il concetto di animazione che verrà ulteriormente chiarito e affrontato in seguito.

210 REM***fine programma*****

500 TIME\$ OFF:MENU

990 REM***legge le note e le suona****

1000 RESTORE

1010 FOR I=1 TO M

1020 READ T:SOUND T,D:SOUND 0,D

1030 READ T:SOUND T,D:SOUND 0,D

1040 NEXT I

1045 REM***suona le ore*****

1050 IF H>12 THEN HT=H-12 ELSE IF H=0 THEN HT=12 ELSE HT=H

1060 IF M=4 THEN FOR I=1 TO HT:SOUND 0,D*2:SOUND C,D:NEXT I

1070 M=M+1:IF M>4 THEN M=1

1080 IF M=4 THEN H=H+1:IF H>23 THEN H=0

1090 REM***analizza la funzione time\$**

1100 IF H<10 THEN H\$="0"+RIGHT\$(STR\$(H),1)

ELSE H\$=RIGHT\$(STR\$(H),2)

1110 IF M=4 THEN M\$="00" ELSE M\$=RIGHT\$(STR\$(M*15),2)

1120 ON TIME\$=H\$+"."+M\$+"00"

GOSUB 1000

1130 RETURN

1990 REM***note della melodia suonata**

2000 DATA 3728,4697,4184,6269,6269,4184,3728,4697

3000 REM***disegna orologio*****

3010 LINE (91,4)-(153,58),1,B

3020 LINE (93,6)-(151,16),1,B

3030 LINE (93,19)-(151,56),1,BF

3040 LINE (95,21)-(100,41),0,BF

3050 LINE (144,21)-(149,41),0,BF

3060 RETURN

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 al negoziante autorizzato Olivetti o al punto vendita Olivetti. Per il noleggio a lungo termine Olivetti. Divisione Personal Computer. Via Marconi 12
 20123 Milano
 TELEFONO