

Spediz. in abbonamento postale GR 11/70 L. 2.200
(...)

63 CORSO PRATICO COL COMPUTER

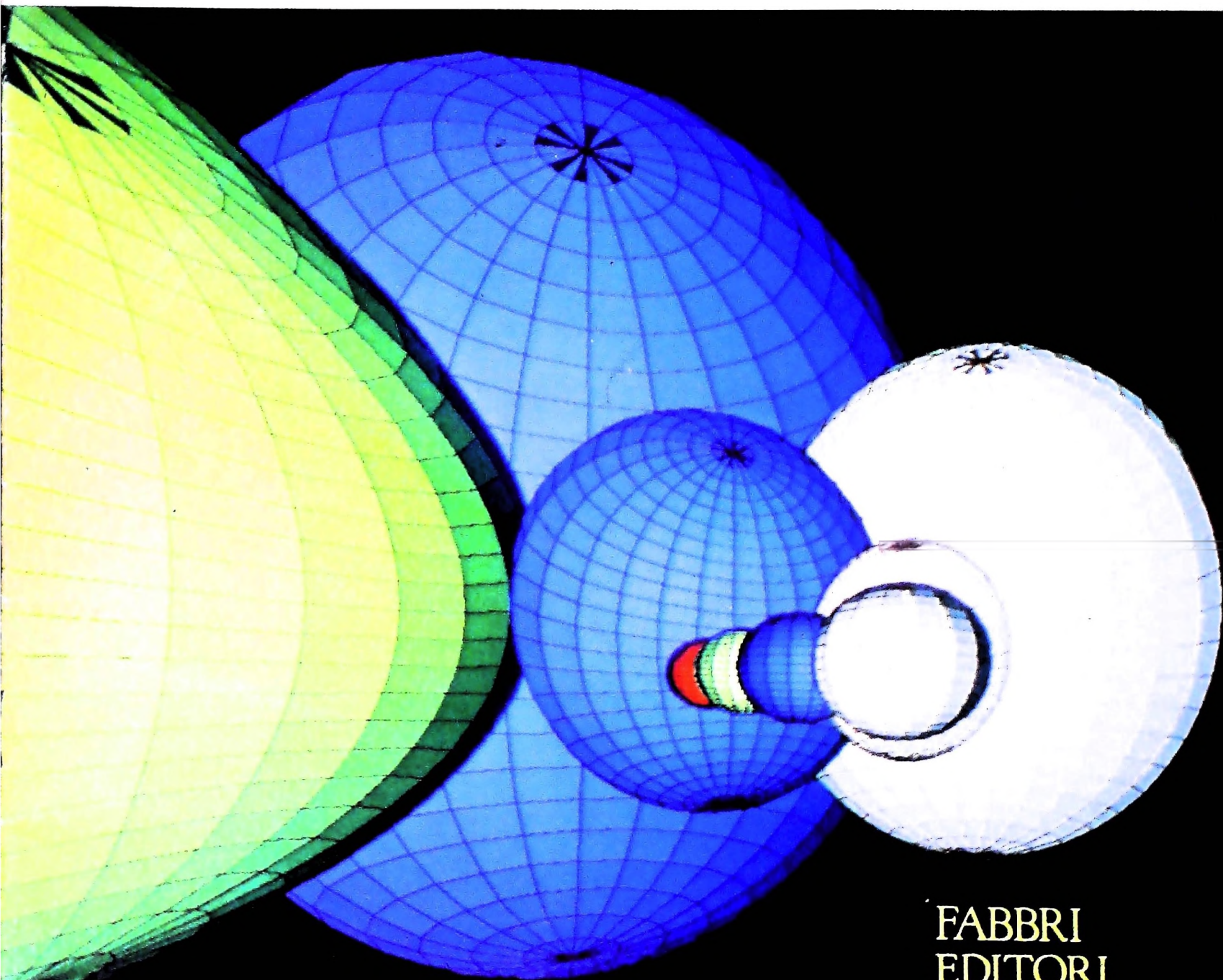
422154

F4 F6 F5 F7
diretto da **GIANNI DEGLI ANTONI**

è una iniziativa
FABBRI EDITORI

in collaborazione con
BANCO DI ROMA

e **OLIVETTI**



**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 trattenute 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
CLAUDIO PARMELLI, MARCO ANELLI, DIEGO BIASI, 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 FENGLI

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

La foto di copertina è una produzione EIDOS

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. 00282, 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. 63 - 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 collettivo da mutate condizioni di mercato.

I PERSONAL ROBOT

Dai racconti di Isaac Asimov alla realtà.

Un calcolatore è una macchina costruita per elaborare dati. Le sue possibilità di interagire con l'ambiente che lo circonda sono quindi limitate allo scambio di informazioni, scambio che avviene tramite le periferiche. Interfacciato con apposite apparecchiature, però, qualsiasi calcolatore può essere messo in grado di manipolare oggetti reali.

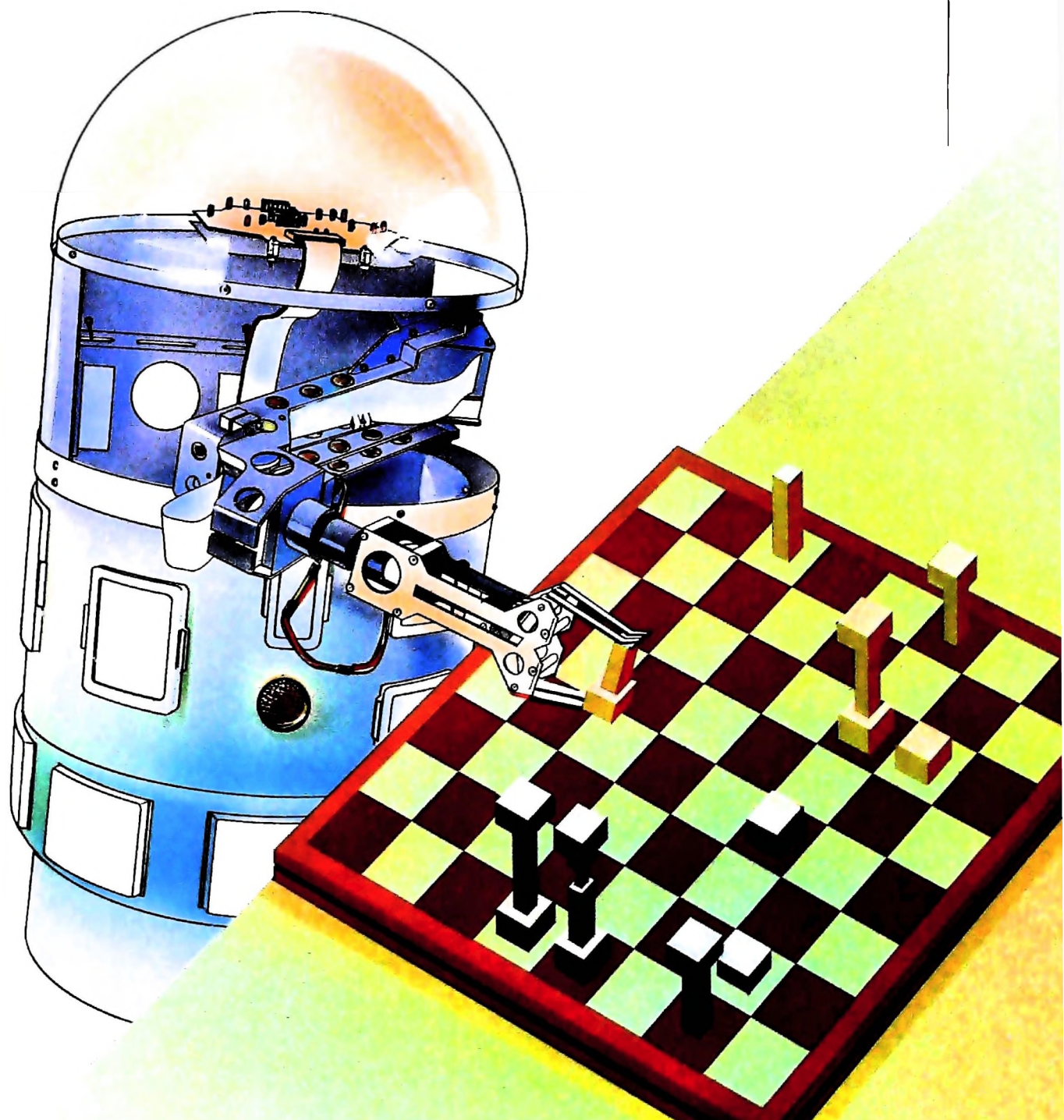
Questo "calcolatore/manipolatore" manterrà ovviamente la caratteristica peculiare di un computer, cioè quella di essere capace, se dotato del programma adatto, di svolgere più

compiti diversi e quindi di distinguersi radicalmente dalle varie apparecchiature automatiche "dedicate" (cioè realizzate per svolgere un solo lavoro) a cui fino ad ora la tecnica ci aveva abituato.

Per fare un esempio molto semplice una macchina di questo tipo potrebbe con la stessa facilità manovrare contenitori di sostanze radioattive, oppure spostare i pezzi su di una scacchiera.

Con un termine notissimo, preso a prestito da una commedia

Ecco una scena che sino a qualche anno fa avremmo detto esser tratta da qualche improbabile film di fantascienza: un robot che gioca a scacchi. Il robot non è altro, in fondo, che un calcolatore dotato di apparecchiature e mezzi meccanici che lo mettono in grado di interagire con l'ambiente, manipolare oggetti, muoversi.



cecoslovacca degli anni Venti, queste macchine dotate della possibilità di manipolare oggetti oppure di spostarsi sono note come "robot".

I robot, per la loro versatilità ed economicità, hanno trovato un vasto impiego nell'industria e nella ricerca scientifica, dove eseguono compiti troppo pericolosi per gli esseri umani, oppure troppo noiosi e ripetitivi.

Macchine di questo genere sono quasi sempre di dimensioni ragguardevoli e hanno prezzi dell'ordine di centinaia di milioni. I fondamenti della robotica (la scienza che studia lo sviluppo e l'impiego dei robot) possono però essere imparati e sfruttati anche da chi possiede un personal computer e a un prezzo che, se non proprio irrisorio, è comunque alla portata di molti.

Robot fissi

I tipi più semplici di robot "domestici" sono costituiti da un braccio meccanico, mosso da una serie di motori passo-passo e collegato a un personal computer. Il braccio meccanico termina con una pinza di prensione che simula una mano.

Le azioni che si vogliono far compiere al robot devono essere scomposte in una serie di movimenti elementari del tipo "avanti 10 centimetri", "ruota di 40 gradi" eccetera, e inserite in un programma.

Alcuni modelli più sofisticati consentono di "mostrare" al robot che cosa deve fare guidandolo con un joystick.

La sequenza dei movimenti viene in questo caso scomposta e memorizzata da un programma apposito residente nel computer, pronta per essere rieseguita oppure modificata.

Robot di questo tipo non presentano particolari problemi costruttivi o di utilizzo, e sono molto usati in laboratori e istituti tecnici per l'insegnamento e l'addestramento all'impiego dei robot industriali.

Una macchina di questo tipo è però semplicemente un "manipolatore", lontana dalle capacità di un essere umano.

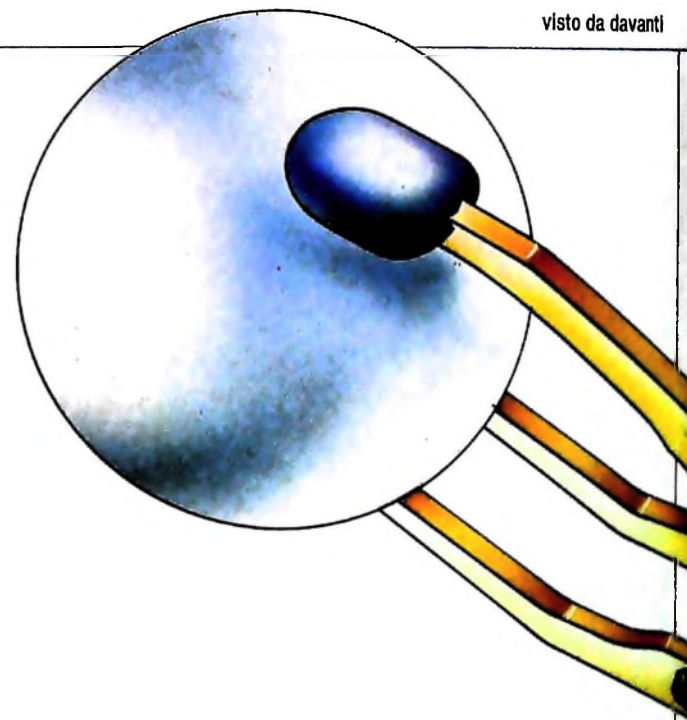
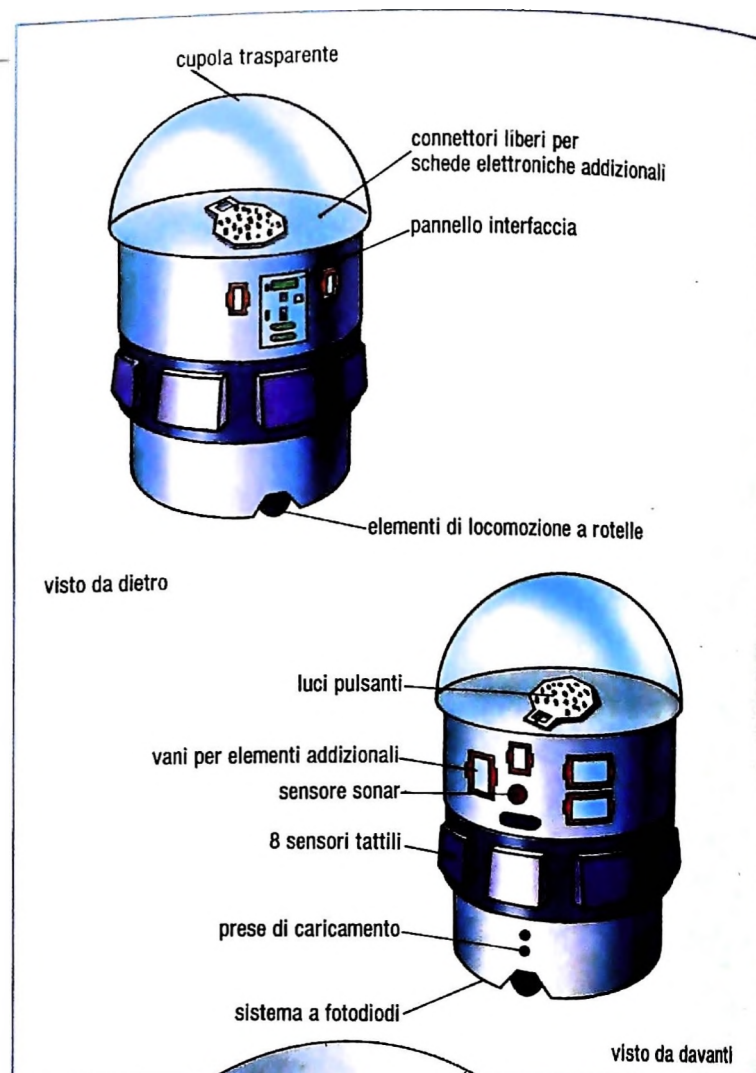
Nel compiere anche il movimento più semplice l'uomo integra, senza rendersene conto, decine di informazioni provenienti da fonti diverse.

Il posizionamento della mano e delle dita è infatti quasi sempre guidato dalla vista, che fornisce tra l'altro dati relativi alle tre dimensioni, mentre alcune strutture particolari, presenti nei muscoli e chiamate "propriocettori", consentono di graduare la forza e di stringere un oggetto pesante, oppure di afferrare un uovo con sicurezza, ma senza romperlo.

L'estendere una percezione sensoriale di questo tipo a un robot, domestico o meno, rappresenta forse il problema più grosso dell'intera robotica.

Alcune soluzioni a questo proposito sono state tentate con cellule fotoelettriche e telecamere, oppure con speciali sensori che mimano l'azione dei propriocettori.

Oltre alle maggiori complicazioni costruttive, questi robot implicano la realizzazione di programmi di controllo molto più complessi, che tengano conto anche delle informazioni provenienti dagli "organi di senso" e graduino di conseguenza le azioni e le reazioni della macchina.



In alto, nel riquadro: parti di un robot. Queste macchine — ed è difficile dire quanto ciò sia nelle intenzioni del costruttore — riescono persino ad assumere un aspetto goffamente umano. Qui a destra: il "braccio" di un robot afferra con delicatezza una pallina.

Robot mobili

Se il motore passo a passo controllato dal computer è collegato a una o più ruote (solitamente tre per motivi di stabilità) si ottiene una macchina in grado di muoversi, cioè un "robot" molto più simile all'idea tradizionale di robot a cui libri e film ci hanno abituato.

Il computer che coordina il tutto può risiedere nel robot stesso oppure, nei modelli più economici, svolgere la propria funzione tramite un collegamento via radio o via filo.

Oltre che della possibilità di muoversi, i modelli più diffusi di robot domestici sono dotati di uno o più tipi di sensori, solitamente a raggi infrarossi, a ultrasuoni e a contatto ("bumper").

Molti di questi robot offrono anche la possibilità di montare un braccio manipolatore, concettualmente simile ai bracci di cui si è parlato prima.

Le possibilità offerte da una macchina di questo tipo accoppiata a un computer sono quasi illimitate. Integrando opportunamente le informazioni sensoriali con i comandi di movimento è infatti possibile creare un programma che simuli il comportamento umano in modo via via sempre più adeguato e si avvicini ai confini dell'intelligenza artificiale.

Seguendo le opportune istruzioni un robot è in grado per esempio di mappare tutti gli ostacoli di un ambiente e quindi di muoversi all'interno di esso con precisione millimetrica, per servire da bere agli ospiti oppure per andare a rispondere alla porta.

Qualora nel suo raggio di azione entri un oggetto in movimento il robot potrà salutare l'oggetto stesso (assumendo che si tratti di un essere umano), oppure farsi da parte per evitare di essere urtato. Una fiamma di dimensioni superiori a un certo limite potrebbe inoltre attivare uno speciale sottoprogramma che farebbe lanciare l'allarme al robot stesso e lo farebbe precipitare allo spegnimento del principio di incendio.

Nonostante l'apparenza fantascientifica tutte queste applicazioni sono già alla portata dei modelli di robot attualmente disponibili sul mercato. Quasi tutti gli esperti di informatica prevedono inoltre che in un futuro abbastanza prossimo moltissimi "computermaniaci" diventeranno "robomaniaci", creando programmi sempre più sofisticati e versatili per i robot che l'industria sta cominciando a sfornare.

Con tutta probabilità, quindi, mentre il personal computer si farà carico dei compiti più ripetitivi e noiosi nel campo della gestione dei dati, il personal robot svolgerà in ambiente domestico e lavorativo compiti altrettanto noiosi e ripetitivi, come la sorveglianza o la pulizia.

Robot simulati

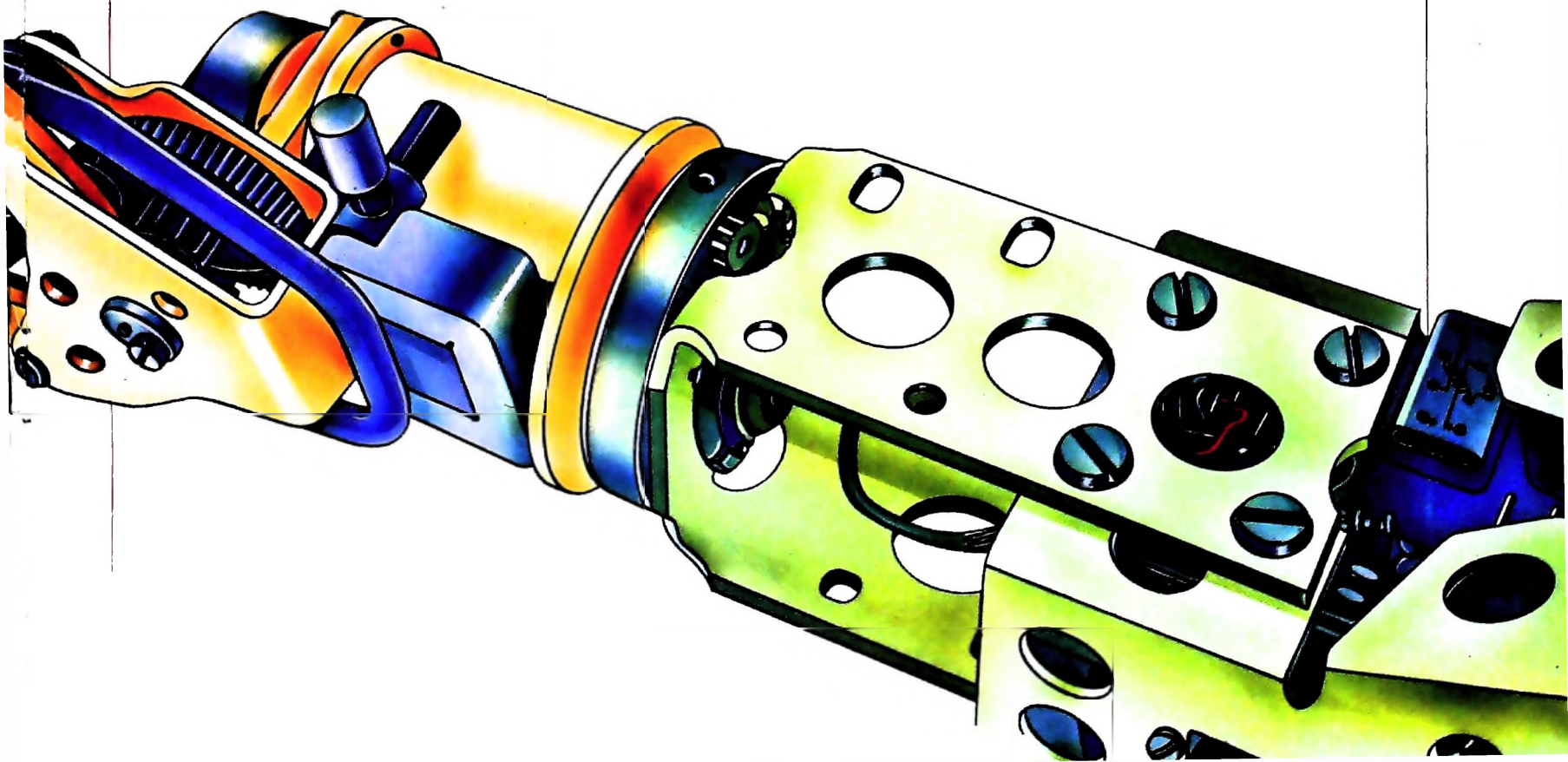
Il costo di un personal robot, da uno ad alcuni milioni, pur non essendo in assoluto altissimo, ne limita tuttavia per il momento la diffusione.

Per motivi didattici e sperimentali è però possibile simulare, all'interno della memoria di un computer, un robot e l'ambiente in cui esso dovrà operare, in modo da poter verificare in tutte le condizioni l'efficacia degli algoritmi di movimento e di controllo che si intendono impiegare.

Il tipo più semplice di "robot simulato" è rappresentato dalla "tartaruga" (turtle) che disegna, nata con il LOGO e ora adottata anche da alcuni dialetti Basic.

Tramite appositi comandi, semplicissimi e primitivi, è infatti possibile far tracciare a questo immaginario animale figure e disegni della complessità voluta.

Altri programmi di simulazione consentono invece di costruire quasi fisicamente il proprio robot e di dotarlo dei tipi di sensore e delle capacità di movimento desiderate, nonché di "programmarlo", in modo da divenire dei proietti robotici anche se non si dispone di un robot vero e proprio.



MEMO-MANAGER

Un programma con funzioni di segreteria
ci permetterà di pianificare i nostri impegni.

07/09/85 19:11 "AULA MAGNA Via Celoria 21" Lezione

04/09/85 8.30 "Studio Avvocato Celli" Pratica per Societa'

04/09/85 14.30 "Riunione Esecutivo" O.d.G.

"Assicurazione Auto"

A prima vista questo programma potrebbe sembrare un doppiopione dello SCHEDL residente in M10. In realtà MEMO-MANAGER non soltanto permette la registrazione di appuntamenti (riunioni, conferenze, lezioni ecc.), ma ne consente anche l'estrazione selettiva, in funzione della data, dell'intervallo di date o dell'argomento.

Utilizzo del programma

Entrambe le routine del menù, Inserimento e Stampa, utilizzano lo stesso criterio di accettazione delle informazioni che sono state digitate.

Nel caso si commettessero errori in fase di digitazione, per correggere un singolo carattere sarà sufficiente utilizzare il tasto BS (backspace) per correggere l'errore. Se invece si volesse annullare l'intera parola premere il tasto funzione F7.

Lanciato il programma, apparirà il menù di servizio.

Per memorizzare un appuntamento nell'apposito file, automaticamente creato da Memo-Manager, APPSCH.DO, premere il tasto funzione F1 per portare il cursore sull'opzione INSERIMENTO APPUNTAMENTI e quindi premere il tasto funzione F2.

Per effetto di tale operazione apparirà la maschera di inserimento che si dovrà riempire secondo le seguenti modalità:

DATA APPUNTAMENTO: inserire la data nella forma mm/gg/aa. Per esempio 04/09/85 per identificare il 9 Aprile 1985.

ORA APPUNTAMENTO: inserire l'ora nel formato 0-24 e cioè inserire 19:11 per identificare le 7:11 PM.

LUOGO: digitare il luogo dell'appuntamento, che può esse-

re l'ufficio di una persona piuttosto che un indirizzo.

Per esempio: "sala riunioni 7° piano", oppure "Aula Magna via Celoria 21".

NOTE: infine inserire le eventuali note che possono, per esempio, identificare il motivo dell'incontro o della riunione o della lezione.

Una volta soddisfatti del record costruito, premere il tasto funzione F1 per registrarlo (in forma compressa) nel file APPSCH.DO). Nel caso non si fosse soddisfatti è possibile cancellarlo, prima di registrarlo, premendo il tasto funzione F2 e ritornando così al menù di servizio.

Per visualizzare o stampare alcuni o tutti i record relativi al file APPSCH:DO, posizionare il cursore sull'opzione STAMPA APPUNTAMENTI, nel menù di servizio, e premere il tasto funzione F2. Questo causerà l'apparizione della maschera di inserimento parametri di ricerca che dovrà essere opportunamente compilata.

STAMPA APPUNTAMENTI PER DATA: per visualizzare un appuntamento, è necessario specificarne la data; in questo caso il programma estrarrà tutti gli appuntamenti compresi tra la data di INIZIO CON DATA e quella inserita nel campo FINE CON DATA. Per limitare la ricerca a un solo giorno le due date dovranno essere coincidenti.

Lasciando vuoto il primo campo, il programma inizierà a stampare dall'inizio del file, mentre lasciando vuoto il secondo campo, il programma stamperà fino alla fine del file.

È possibile selezionare gli appuntamenti per specificazioni; per esempio si possono estrarre tutti gli appuntamenti inerenti a conferenze e, se si volesse restringere ancora, inerenti a conferenze di geologia.

Per ottenere tale risultato, digitare la parola nel campo SPE-

CIFICAZIONI seguito dal carattere ottenuto premendo SHIFT GRPH C.

Nel caso si volesse definire una seconda specifica (nell'esempio esposto geologia), digitarla dopo il carattere speciale seguita anch'essa da SHIFT GRPH C e quindi ENTER. Se non si vuole la seconda specifica, dopo il carattere speciale premere ENTER.

Infine, rispondendo opportunamente alla domanda SU SCHERMO O STAMPANTE, si otterrà l'elaborato sul dispositivo voluto.

Il programma

Memo-Manager è composto di 13 moduli che governano le operazioni di input/output e ricerca del programma.

MODULO 1 - Linee 20-210.

È la sezione di inizializzazione e generazione del menù di servizio. Definisce i tasti funzione, regola le funzioni di negativo/positivo del video e limita il numero di file da utilizzare.

In questa fase, con il tasto funzione F8, si ha la rinormalizzazione dei tasti funzione e il ritorno al menù generale.

MODULO 2 - Linee 220-620.

Permette l'inserimento dei dati, richiamando il modulo 7, e la memorizzazione dei record nel file APPSCH.DO.

Dopo l'inserimento dei dati relativi a un appuntamento, la subroutine legge il file record per record e confronta le relative date e ore con il record appena inserito.

Se il nuovo record ha data e ora inferiori a quello appena letto, la subroutine scrive il nuovo record in un file temporaneo (TMPSCH.DO) prima di scriverci il vecchio. In caso contrario la routine scrive in TMPSCH.DO solo il vecchio record. (In caso di eguaglianza, l'ultimo record viene assunto come maggiore.) Al termine del processo di inserimento, il modulo cancella il file APPSCH.DO e rinomina TMPSCH.DO come APPSCH.DO.

MODULO 3 - Linee 630-1100.

Permette la visualizzazione o la stampa di una parte o tutto l'archivio APPSCH.DO, per mezzo delle specifiche delle date di inizio e fine nonché dei termini di specificazione.

MODULO 4,5 - Linee 1100-1450

Permettono, nell'ordine, la stampa in formato 80 colonne e la visualizzazione su display a cristalli liquidi.

MODULO 6, 7, 8, 9, 10, 11, 12, 13 - Linee 1460-2240.

Permettono il controllo sui tasti premuti e sui caratteri da visualizzare.

```

10 ' *****
11 '          MEMO MANAGER
12 '
13 '          by C.V.P.
14 ' *****
15 '
16 RV$=CHR$(27)+"p"
   :NV$=CHR$(27)+"q"
17 CLS
   :LINE(32,5)-(207,18),1,BF
18 PRINT$54,RV$;"Memo Manager";NV$
19 LINE(29,2)-(210,21),1,B
20 PRINT$204,"Premi un tasto per continuare..."
21 M$=INKEY$
   :IFM$=""THEN21
22 REM * MENU GENERALE *
30 CLEAR 500
40 MAXFILES=2: DEFINT A-Z
50 RV$=CHR$(27) + "p": NV$=CHR$(27) + "q"
60 KEY1,"N": KEY2,"S": KEY3,"": KEY4,"": KEY5,"":
   KEY6,"": KEY7,CHR$(27): KEY8,"M"
70 CLS: PRINT NV$;
80 PRINT$8,"Benvenuti al Memo Manager"
90 PRINT$280,"Next Selez.";
100 PRINT$315,"Menu";
110 OP$(0)="Inserimento Appuntamenti"
   :PP(0)=88
120 OP$(1)="Stampa Appuntamenti"
   :PP(1)=128
130 R=0
140 FORI=0TO1

```

```

:PRINT$PP(I),NV$;OP$(I);
:NEXT I
150 PRINT$PP(R),RV$;OP$(R);NV$;
160 AS=INKEY$: IF AS="" THEN 160
170 IF AS="N" THEN R=(R+1)MOD2: GOTO 140
180 IFA$="M" THEN KEY1,"Files"+CHR$(13)
:KEY2,"Load"+CHR$(34)
:KEY3,"Save"+CHR$(34)
:KEY4,"Run"+CHR$(13)
:KEY8,"Menu"+CHR$(13):MENU
190 IF AS <> "S" THEN BEEP: GOTO 140
200 ON R+1 GOSUB 220,630,
210 GOTO 60
220 REM * INSERIMENTO APPUNTAMENTI *
230 ON ERROR GOTO 590
240 CLS: PRINT NV$;
250 PRINT$0,"Data Appuntamento: _/_/_";
260 PRINT$40,"Ora Appuntamento: _:_";
270 PRINT$80,"Luogo: "+STRING$(30,"_")
280 PRINT$120,"Note: "+STRING$(30,"_");
290 PRINT$280,"Rcrd Cncl";
300 BFS=STRING$(70," ")
310 CC(0)=1: PP(0)=20
320 CC(1)=1: PP(1)=23
330 CC(2)=1: PP(2)=26
340 CC(3)=1: PP(3)=58
350 CC(4)=1: PP(4)=61
360 CC(5)=29: PP(5)=90
370 CC(6)=29: PP(6)=130
380 MW=6: MB=69: WD=0: BP=0: CH=0
390 GOSUB 1620
400 IF C$=CHR$(2) GOTO 560
410 OPEN "APPSCH" FOR INPUT AS 1
420 OPEN "TMPSCH" FOR OUTPUT AS 2
430 S1$=MID$(BFS,5,2) + MID$(BFS,1,4) + MID$(BFS,7,4)
440 IF EOF(1) THEN PRINT#2,BFS;: GOTO 520
450 IP$=INPUT$(70,1)
460 S2$=MID$(IP$,5,2) + MID$(IP$,1,4) + MID$(IP$,7,4)
470 IF S1$ < S2$ THEN PRINT#2,BFS;: PRINT #2,IP$;ELSE
PRINT#2,IP$;: GOTO 440
480 IF EOF(1) GOTO 520
490 IP$=INPUT$(70,1)
500 PRINT#2,IP$;
510 GOTO 480
520 CLOSE 1,2
530 KILL "APPSCH.DO"
540 NAME "TMPSCH.DO" AS "APPSCH.DO"
550 GOTO 580
560 PRINT$292,NV$;"Operazione annullata";
570 FOR I=1 TO 500 : NEXT I
580 RETURN
590 IF ERR|_52 THEN PRINT$240,"Errore nr. ";ERR;" in linea ";ERL;
:STOP
600 OPEN "APPSCH.DO" FOR OUTPUT AS 1
610 CLOSE
620 RESUME

```



```

630 REM * STAMPA APPUNTAMENTI *
640 ON ERROR GOTO 590
650 CC(0)=1: PP(0)=20
660 CC(1)=1: PP(1)=23
670 CC(2)=1: PP(2)=26
680 CC(3)=1: PP(3)=58
690 CC(4)=1: PP(4)=61
700 CC(5)=1: PP(5)=64
710 CC(6)=20: PP(6)=99
720 CC(7)=0: PP(7)=154
730 CLS: PRINT NV$;
740 PRINT$0,"Inizio con Data: _/_/_";
750 PRINT$40,"Fine con Data: _/_/_";
760 PRINT$80,"Specificazioni: "+STRING$(22," ");
770 PRINT$120,"Su Schermo o Stampante (S/P):_";
780 PRINT$280,"Print Cncl";
790 KEY1,"N": KEY2,"C": KEY8,""
800 BF$=STRING$(34," ")
810 BP=0: CH=0: WD=0: MB=33: MW=7
820 GOSUB 1620
830 IF C$=CHR$(2)THEN PRINT$292,"Operazione annullata"
;:GOTO1090
840 SD$=MID$(BF$,5,2) + MID$(BF$,1,4)
850 ED$=MID$(BF$,11,2) + MID$(BF$,7,4)
860 IF ED$=" " THEN ED$="991231"
870 TC=0
880 TP$(1)="": TP$(2)="
890 PT=INSTR(13,BF$,CHR$(255))
900 IF PT=0 THEN 1050
910 TP$(1)=MID$(BF$,13,PT-13)
920 FOR I=1 TO LEN(TP$(1))
930 IF MID$(TP$(1),I,1) < "a" OR MID$(TP$(1),I,1) > "z"
THEN 950
940 MID$(TP$(1),I,1)=CHR$(ASC(MID$(TP$(1),I,1))-32)
950 NEXT I
960 NP=PT+1: TC=1
970 PT=INSTR(NP,BF$,CHR$(255))
980 IF PT=0 THEN 1050
990 TP$(2)=MID$(BF$,NP,PT-NP)
1000 FOR I=1 TO LEN(TP$(2))
1010 IF MID$(TP$(2),I,1) < "a" OR MID$(TP$(2),I,1) >
"z" THEN 1030
1020 MID$(TP$(2),I,1)=CHR$(ASC(MID$(TP$(2),I,1))-32)
1030 NEXT I
1040 TC=2
1050 OPEN "APPSCH" FOR INPUT AS 1
1060 IF MID$(BF$,34,1)= "P" OR MID$(BF$,34,1)= "p"THEN
GOSUB 1110 ELSE GOSUB 1290
1070 CLOSE
1080 PRINT$292,NV$;"FINE FILE";
1090 FOR I=1 TO 1000: NEXT I
1100 RETURN
1110 REM * STAMPA SU STAMPANTE *
1120 OPEN "LPT:" FOR OUTPUT AS 2
1130 CR$=CHR$(13): FF$=CHR$(12)
1140 PRINT$2,CR$;CR$;TAB(33);"Appuntamenti";CR$

```

```

1150 PRINT#2," Data Ora Luogo";TAB(27);"Note"
1160 LC=6
1170 OD$=""
1180 GOSUB 1460
1190 IF PF=0 THEN 1280
1200 ND$=MID$(BF$,1,2) + "/" + MID$(BF$,3,2) + "/" +
MID$(BF$,5,2)
1210 IF ND$=OD$ THEN ND$="" " ELSE OD$=ND$:
LC=LC+1: PRINT#2,""
1220 PRINT#2,ND$;
1230 PRINT#2," " + MID$(BF$,7,2) + ":" + MID$(BF$,9,2)
+ " ";
1240 PRINT#2,MID$(BF$,11,30);" ";
1250 PRINT#2,MID$(BF$,41,33)
1260 LC=LC+1
1270 IF LC < 60 THEN PRINT FF$: GOTO 1180 ELSE 1140
1280 RETURN
1290 REM * STAMPA SU VIDEO *
1300 CLS: PRINT NV$;
1310 KEY1,"N": KEY2,"Q"
1320 PRINT"Data Appuntamento:"
1330 PRINT"Ora Appuntamento:"
1340 PRINT"Luogo:"
1350 PRINT>Note:"
1360 PRINT$280,"Next Stop";
1370 GOSUB 1460
1380 IF PF=0 THEN PRINT$292,"FINE FILE";:GOTO1450
1390 PRINT@ 18,LEFT$(BF$,2) + "/" + MID$(BF$,3,2) + "/"
+ MID$(BF$,5,2);
1400 PRINT@ 58,MID$(BF$,7,2) + ":" + MID$(BF$,9,2);
1410 PRINT@ 90,MID$(BF$,11,30);
1420 PRINT@ 130,MID$(BF$,41,33);
1430 AS=INKEY$: IF AS="N" THEN 1370
1440 IF AS <> "Q" THEN 1430
1450 RETURN
1460 REM * CONFERMA INPUT *
1470 IF EOF(1) THEN PF=0: GOTO 1610
1480 BF$=INPUT$(70,1): PF=1
1490 BD$=MID$(BF$,5,2) + MID$(BF$,1,4)
1500 IF BD$ < SD$ OR BD$ > ED$ THEN 1460
1510 IF TC=0 GOTO 1610
1520 TB$=BF$
1530 FOR I=11 TO 83
1540 LC$=MID$(TB$,I,1)
1550 IF LC$ < "a" OR TC$ > "z" THEN 1570
1560 MID$(TB$,I,1)=CHR$(ASC(LC$)-32)
1570 NEXT I
1580 FOR I=1 TO TC
1590 IF INSTR(1,TB$,TP$(I))=0 THEN 1460
1600 NEXT I
1610 RETURN
1620 REM * ROUTINE INPUT *
1630 KEY1,CHR$(1): KEY2,CHR$(2): KEY8,""
1640 AS$=MID$(BF$,BP+1,1)
1650 IF AS$="" THEN AS$=""
1660 PRINT$PP(WD)+CH,RV$:AS$;
1670 CS=INKEY$: IF CS="" THEN 1670

```

Lezione 62

Altri formati di stampa

Completiamo in questa lezione la nostra carrellata sui formati di stampa. L'ultimo esempio che avevamo visto corrispondeva alla stampa di una sequenza di numeri e della relativa somma, ed era stato collegato a stampe di rapporti finanziari, in cui il segno o l'evidenza del segno "-" sono spesso messi alla fine del numero. Sempre riferendoci a quell'esempio, esaminiamo il formato

"\$"

che, anteposto al normale formato numerico, permette di far precedere la stampa dal simbolo "\$". Così, il programma:

```

10 REM Somma di N numeri
20 READ N
30 LPRINT "Somma di ";N;" valori"
40 LPRINT
45 LET S=0
50 FOR I=1 TO N
60 READ X
70 LET S=S+X
75 LPRINT TAB(10);
80 LPRINT USING "$$#####-";X
90 NEXT I
100 LPRINT
110 LPRINT "Somma:";
115 LPRINT TAB(9);
120 LPRINT USING "$$#####-";S
200 DATA 10
210 DATA 12,-45,76,22,-44,-53,4,-9,10,9

```

permette di ottenere la seguente stampa:

Somma di 10 valori

```

$12
$45-
$76
$22
$44-
$53-
$4
$9-
$10
$9

```

Somma: \$18-

Ci sono dei vincoli nell'uso di questo tipo di formato: non è possibile il suo uso con valori in forma esponenziale, né con valori in cui sia richiesto il segno, a meno che

questo non sia collocato alla fine del numero. È evidente infatti l'orientamento di questo formato ad applicazioni finanziarie. Il formato così definito risulta ampliato di due caratteri aggiuntivi, corrispondenti proprio ai due "\$". Il simbolo "\$" è privilegiato nel trattamento di formati di stampa, ma noi possiamo inserire anche le nostre Lire nel formato. Infatti, se facciamo precedere il formato numerico da una stringa (che non contenga, ovviamente caratteri confondibili con quelli di un formato), questa verrà stampata inalterata prima dell'inizio del campo numerico. Consideriamo la modifica del programma precedente:

```
80 LPRINT USING "Lit #####-";X
120 LPRINT USING "Lit #####-";S
```

Tale programma fornisce il seguente risultato:

Somma di 10 valori

```
Lit 12
Lit 45-
Lit 76
Lit 22
Lit 44-
Lit 53-
Lit 4
Lit 9-
Lit 10
Lit 9
```

Somma: Lit 18-

Tale comportamento non è limitato alle stringhe che precedono il formato numerico, ma vale anche per quelle che lo seguono. Un altro formato riguarda la possibilità di riempire gli spazi che restano liberi prima del numero con dei caratteri "*". Si tratta di far precedere il formato numerico da

“**”

Per esempio, usando il formato

“**####-”

nel programma precedente, otteniamo:

Somma di 10 valori

```
Lit ****12
Lit ****45-
Lit ****76
Lit ****22
Lit ****44-
Lit ****53-
Lit ****4
Lit ****9-
Lit ****9-
```

```

Lit ****10
Lit *****9

```

```
Somma: Lit *****18-
```

Questo formato può essere fatto seguire dal "\$\$", combinandone così gli effetti. Ancora, supponiamo di cambiare la linea dei dati del nostro programma, inserendo valori molto grandi, e ampliando le dimensioni dei campi da visualizzare:

```

80 LPRINT USING "Lit ?????????-";X
120 LPRINT USING "Lit ?????????-";S
200 DATA 10
210 DATA 12E3,-45E4,76E6,22E5,-44E6,-53E4,4E3,
      -9E5,10E6,9E6

```

Il risultato

```
Somma di 10 valori
```

```

Lit 12000
Lit 450000-
Lit 76000000
Lit 2200000
Lit 44000000-
Lit 530000-
Lit 4000
Lit 900000-
Lit 10000000
Lit 9000000

```

```
Somma: Lit 51336000
```

non garantisce un'adeguata leggibilità, in quanto non riusciamo a distinguere chiaramente il numero di zeri che segue ogni valore. In genere siamo abituati, proprio per migliorare la leggibilità, a dividere il numero in terzetti di cifre, a partire dal fondo, per evidenziare le migliaia, i milioni, i miliardi, e così via.

L'abitudine italiana è quella di usare la virgola come elemento di separazione tra parte e parte decimale, e il punto come elemento di separazione di ogni terzetto; la notazione anglosassone usa la convenzione opposta, e i calcolatori si adeguano a quest'ultima. Così, il valore che noi italiani scriveremmo come 1.050.234,56 (corrispondente) a unmilionecinquantamiladuecentotrentaquattro e 56 centesimi), viene scritto con notazione anglosassone nella forma

1,050,234.56

I comandi di formato mettono a disposizione tale possibilità di specificare i numeri con i terzetti separati da "," mettendo tale carattere subito prima del punto decimale nel comando che descrive la struttura del numero.

Rifacciamoci al nostro programma: poiché esso non prevede la stampa di parti decimali, inseriremo la virgola alla fine degli indicatori del numero di cifre, subito prima del segno; a causa della presenza delle virgole, l'ampiezza del campo dovrà essere significativamente aumentata. Il nostro programma diviene allora:

```
80 LPRINT USING "Lit  EEEEEEEEEEEE,-";X
120 LPRINT USING "Lit  EEEEEEEEEEEE,-";S
```

e produce il risultato

Somma di 10 valori

```

Lit      12,000
Lit      450,000-
Lit      76,000,000
Lit      2,200,000
Lit      44,000,000-
Lit      530,000-
Lit       4,000
Lit      900,000-
Lit     10,000,000
Lit      9,000,000
```

Somma: Lit 51,336,000

I comandi di vari formati possono essere giustapposti; così l'istruzione

```
10 PRINT USING "A£BÇÇC£+D"; 1, "QQ", 2
```

vede la giustapposizione di tre comandi per stampa di numeri e stringhe, con caratteri da visualizzare inalterati inframmezzati. Il risultato di tale stampa sarebbe:

A1BQQC2+D

Cosa abbiamo imparato

In questa lezione abbiamo visto:

- il formato "\$\$" per far precedere valori numerici dal simbolo "\$";
- la possibilità di inserire stringhe da stampare inalterate nel comando di formato;
- il comando "*" per riempire di asterischi gli spazi che precedono un valore numerico;
- il comando "," per separare i terzetti delle migliaia, dei milioni ecc. nella stampa di un numero;
- la costruzione di comandi di formato complessi mediante la giustapposizione di più comandi semplici.

```

1680 IF C$=CHR$(28) THEN GOSUB 1840
1690 IF C$=CHR$(29) THEN GOSUB 1760
1700 IF C$ > CHR$(31) THEN GOSUB 2040
1710 IF C$=CHR$(8) THEN GOSUB 2150
1720 IF C$=CHR$(27) THEN GOSUB 1920
1730 IF C$=CHR$(13) THEN GOSUB 1980
1740 IF C$=CHR$(1) OR C$=CHR$(2) THEN RETURN
1750 GOTO 1640
1760 REM * CURSORE A SINISTRA *
1770 PRINT$PP(WD)+CH,NV$;AS$;
1780 CH=CH-1: BP=BP-1
1790 IF CH >= 0 THEN GOTO 1830
1800 WD=WD-1
1810 IF WD < 0 THEN WD=MW: BP=MB
1820 CH=CC(WD)
1830 RETURN
1840 REM * CURSORE A DESTRA *
1850 PRINT$PP(WD)+CH,NV$;AS$;
1860 CH=CH+1: BP=BP+1
1870 IF CH <= CC(WD) GOTO 1910
1880 WD=WD+1
1890 IF WD > MW THEN WD=0: BP=0
1900 CH=0
1910 RETURN
1920 REM * ESCAPE*
1930 BP=BP-CH
1940 CH=0
1950 MID$(BFS,BP+1,CC(WD)+1)=STRING$(CC(WD)+1," ")
1960 PRINT$PP(WD)+CH,NV$:STRING$(CC(WD)+1,"_")
1970 RETURN
1980 REM * ENTER*
1990 PRINT$PP(WD)+CH,NV$;AS$;
2000 BP=BP+(CC(WD)-CH)+1: WD=WD+1
2010 IF WD > MW THEN WD=0: BP=0
2020 CH=0
2030 RETURN
2040 REM * LETTERA STAMPABILE.*
2050 MID$(BFS,BP+1,1)=C$
2060 BP=BP+1
2070 IF C$=" " THEN C$=""
2080 PRINT$PP(WD)+CH,NV$;C$;
2090 CH=CH+1
2100 IF CH <= CC(WD) GOTO 2140
2110 WD=WD+1
2120 IF WD > MW THEN WD=0: BP=0
2130 CH=0
2140 RETURN
2150 REM * BACKSPACE *
2160 PRINT$PP(WD)+CH,NV$;AS$;
2170 BP=BP-1
2180 CH=CH-1
2190 IF CH >= 0 GOTO 2230
2200 WD=WD-1
2210 IF WD < 0 THEN WD=MW: BP=MB
2220 CH=CC(WD)
2230 MID$(BFS,BP+1,1)=""
2240 RETURN

```

I GIOCHI-RACCONTO

All-texte-game, ovvero una maniera per dialogare con la macchina più moderna attraverso un antico strumento di comunicazione.

Ecco i giochi (Murder game e Adventur game) attraverso cui si compie la vendetta di Gutenberg; questi game infatti si basano sul concetto di letteratura interattiva: il computer propone delle istruzioni o dei frammenti narrativi, il giocatore deve saperli interpretare e fornire le opportune istruzioni. Se il calcolatore giudica soddisfacenti le risposte, allora il gioco-racconto procede. E perché Gutenberg? Perché questi giochi si basano sostanzialmente sulla scrittura, e non sulla grafica. Il best-seller che sta trionfando nelle librerie degli Stati Uniti è un "giallo" con tanto di morti, ispettori di polizia, misteri irrisolti. Come mai piace tanto? Piace perché non è un thriller da leggere ma da far "girare" sul personal computer. All'atto dell'acquisto, al lettore giocatore vengono infatti consegnati un manuale di istruzioni, un floppy disk che racchiude la trama e dodici ore di tempo per decifrare il messaggio. Basta con le guerre stellari, basta con "Pitfall" o con "Pac Man"; adesso è venuto il momento di "Deadline", una delle tante storie "pubblicate" dalla Infocom di Cambridge, Massachusetts, una casa editrice che invece della carta usa supporti magnetici.

Le cose tuttavia non sono così semplici. È vero che il computer restituisce una storia a cui non manca nessuno degli ingredienti classici del "giallo": paesaggi tenebrosi, descrizioni dettagliate di persone, frammenti di dialogo disseminati di indizi; e intanto i sospetti si accavallano, elementi significativi appaiono e scompaiono sullo schermo, insospettabili impiegati mentono e uccidono ancora. Ma per risolvere il quesito finale ci vogliono ore e ore di fatica computeristica. Chiave di volta è il cosiddetto "analizzatore di parole", quel programma che interpreta gli ordini della tastiera. All'inizio, questi analizzatori accettavano solo ordini di una o due parole ("segui X", "uccidi Y") e il racconto si snodava faticosamente. Adesso, grazie a tecniche di intelligenza artificiale, le frasi si articolano attraverso aggettivi, preposizioni e coniugazioni di verbi, tipo "Prendi la bomba e mettila nel cassetto" oppure "Sei completamente fuori strada!". Giudicate secondo il metro letterario, le storie della Infocom appaiono leggermente zoppicanti, i personaggi scolpiti un po' con l'accetta, lo stile non eccelso. Tuttavia rappresentano una svolta storica sia per quel che riguarda il mondo dei computer game sia per quel che riguarda, più in generale, il rapporto letteratura/informatica.

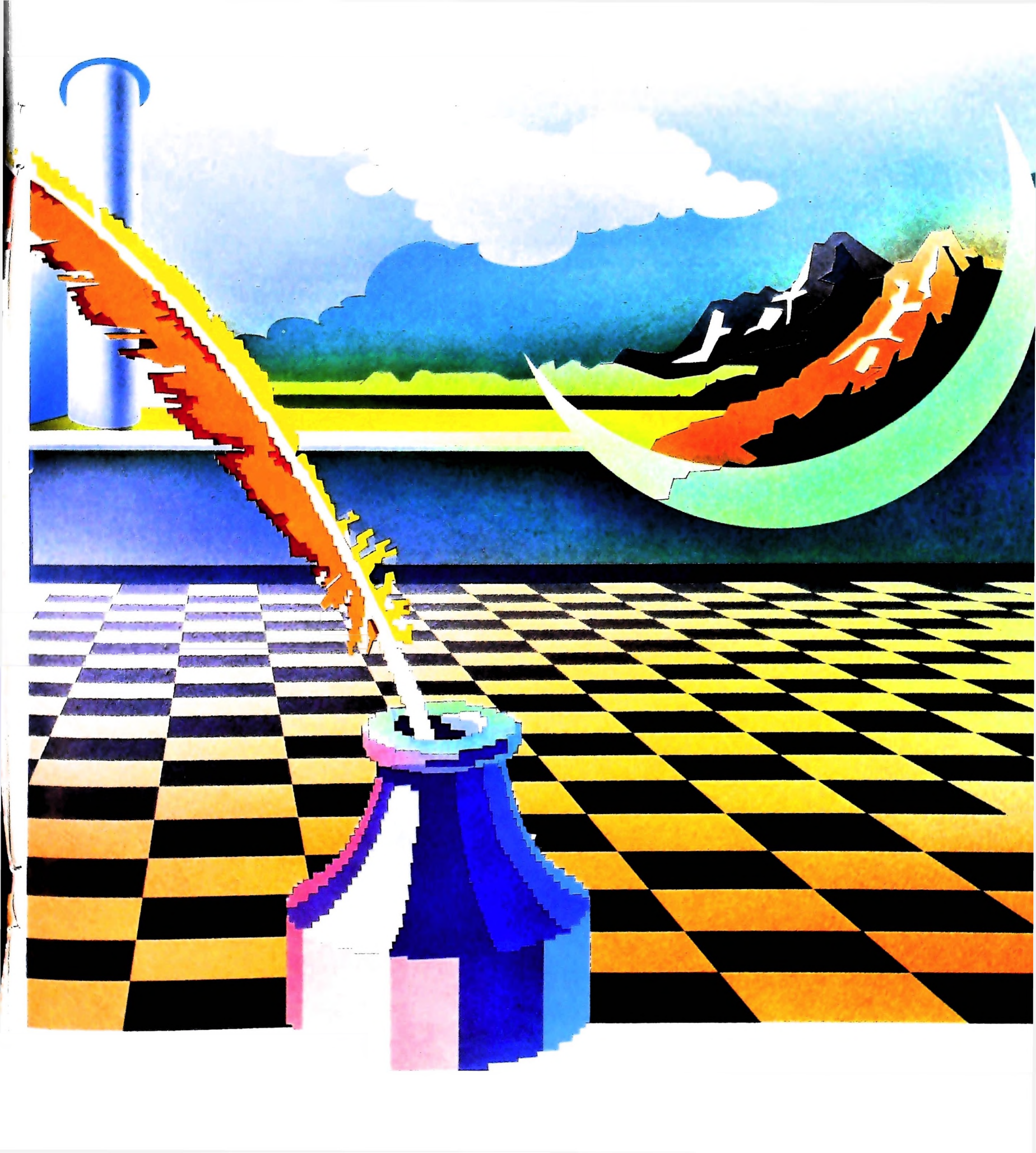
Per comprendere meglio come funzionino questi giochi dia-

mo qui di seguito l'inizio di alcuni dei più celebri giochi "all-texte-game":

"Deadline": il signor Robner è stato assassinato, anche se la morte sembra un suicidio. Io sospetto che sia stato il figlio, anche se non ne ho le prove. Dipende tutto dal movente. Io ho seguito il figlio e l'ho scoperto mentre entrava in una stanza segreta. Sono sicuro che il movente è da ricercarsi in quella stanza segreta. Ma perché il giardiniere continua ad andare avanti e indietro con la scala? E che significa quella telefonata? Esamina la parete sud...;

"The Witness": in un freddo venerdì sera di febbraio, nel

Gli all-texte-game, della Infocom, rappresentano un anello di congiunzione fra il mondo dei computer game e la letteratura. Il genere letterario che sembra essere più congeniale a questo tipo di game è il "giallo", per la sua struttura ricca di azioni e di colpi di scena.



1938, a Los Angeles un uomo viene ucciso. E voi siete testimoni di questo efferato delitto. Un intrigo alla Chandler. Provate a scoprire l'assassino...;

"Murder on the Zinderneuf": l'azione si svolge su uno Zeppelin che attraversa l'Oceano Atlantico nel 1936. È stato commesso un delitto e ora ben nove persone attendono di essere interrogate. Per cercare gli indizi bisogna manovrare il detective attorno alla stanza finché il computer non annuncia che è inciampato contro qualcosa di interessante...;

"Murder by the dozen": ci sono dodici crimini da decifrare. Alle indagini possono partecipare quattro detective. Benvenuti a casa del maestro del mistero e... scegli il caso che vuoi risolvere...;

"Infidel": antiche scritture in un gioco d'avventura? Sì, anche se si tratta di iscrizioni murarie, di geroglifici egiziani. La loro decifrazione è fondamentale per risolvere gli enigmi della 'Piramide scomparsa', una volta che l'avrete ritrovata, s'intende. La piramide è ricca di trappole mortali costruite dai sacerdoti del tempio per impedire a ladri come voi di raggiungere la tomba della regina.

Giorno dopo giorno Sherlock Holmes passa il tempo seduto alla tastiera del suo personal; smaschera continuamente nuovi criminali ma intanto ha scoperto che può anche comporre, per gioco, poesie. È tutto così semplice, anzi "elementare, Watson".

Analizziamo un adventure-game

Un adventure-game può essere caratterizzato, dal punto di vista implementativo, principalmente da due elementi:

- tecnica programmatica abbastanza semplice
- struttura narrativa particolarmente complicata

È cioè il tessuto narrativo, con la descrizione dei personaggi, degli ambienti, degli oggetti, degli obiettivi che differenzia principalmente i vari giochi. Ciò fa sì che la creazione di un "piccolo" adventure-game non sia fattibile in quanto la definizione della struttura narrativa richiede sempre una certa quantità di spazio. Per comprendere la struttura generale di un adventure-game è quindi necessario osservare i giochi presenti sul mercato e considerare analogie e differenze.

Analizzando la struttura dei comandi, che corrisponde al modo con cui il giocatore interagisce con il gioco, si può dire che i principali adventure-game sono ristretti a quella che viene chiamata struttura dei comandi "TAKE STAFF"; in questo caso il comando è composto esclusivamente da un verbo e da un oggetto:

```
VAI A NORD
ENTRA nella FORESTA
PRENDI l'ANELLO MAGICO
```

Alcuni giochi più sofisticati sono ora in grado di permettere una sintassi dei comandi ben più complicata. Vediamo per esempio la frase:

```
BRUCIA TUTTI I LIBRI
TRANNE QUELLO NERO
```

in cui la struttura sintattica pone un comando e, contemporaneamente, ne ammette un'eccezione.

La trama del gioco, comunque, costituisce dal punto di vista del giocatore il principale discriminante fra i giochi; però trama diversa non implica necessariamente programma diverso.

Ed è proprio questa considerazione che ha spinto talune persone (per esempio Scott Adams) a considerare l'idea di scrivere un "INTERPRETE di AVVENTURE", e cioè un programma generalizzato che accetti in input una trama ed i possibili narrativi, e, a seconda dei comandi introdotti dall'utente, faccia evolvere la storia.

Questo programma deve fornire le procedure generali (che andranno caratterizzate a seconda del gioco) per descrivere un adventure-game:

- muoversi da una stanza all'altra
- memorizzare la descrizione di una stanza
- gestire l'input dei comandi.

La gestione dell'input è forse la parte più delicata del programma in quanto deve essere in grado di fare una serie di controlli che dipendono sia dal gioco in generale, sia dallo stato corrente di esecuzione del programma. Consideriamo per esempio la possibilità di porre un comando del tipo:

```
AFFERRA LA SPADA
```

Il sistema prima deve controllare che il verbo **AFFERRA** sia definito nel sistema (e cioè il sistema è in grado di eseguire l'azione corrispondente all'afferrare), successivamente che **SPADA** sia un oggetto definito nel programma ed infine che **SPADA** si trovi nella stanza in cui si trova il giocatore. È possibile ovviamente ampliare questo schema introducendo controlli sempre più sofisticati (per esempio la **SPADA** è sì nella stanza, ma la stanza è buia e il giocatore non ha una torcia per cui non la può vedere).

Si tratta, peraltro, di problemi assai interessanti da un punto di vista logico.

UNIX (IV)

Proseguiamo l'analisi di Unix, e vediamo le classi dei comandi accessibili per l'utente tramite la shell.

I comandi generali di Unix sono quei comandi che svolgono funzioni di carattere generale; tra questi:

who fornisce l'elenco degli utenti collegati in un certo momento al sistema; è utile quando, con altri comandi, si vuole comunicare con gli altri utenti.

date fornisce la data corrente.

I comandi per la gestione dei file

È l'insieme dei comandi che consentono di trattare i file e le direttrici del file-system:

pwd: (print working directory) fornisce il path-name della direttrice sulla quale si è posizionati;

cd: (change directory) modifica la direttrice corrente permettendo di muoversi all'interno dell'albero del file-system;

ls: (list) consente di conoscere il contenuto delle direttrici; è dotato di numerose opzioni grazie alle quali si ottengono diverse informazioni aggiuntive;

mkdir: (make directory) consente di creare le direttrici che si vogliono utilizzare; così l'albero del file-system "cresce";

rmdir: (remove directory) cancella le direttrici specificate, purché siano vuote (purché, cioè, non contengano file);

cat: (concatenate) concatena il contenuto di più file in uno solo; serve per generare file partendo dal conte-



L'editor "vi": si tratta di un editor interattivo che mostra sul video una pagina o finestra contenente una porzione di testo. Dalla posizione del cursore, contenuto nella pagina video, si possono cancellare, sostituire, aggiungere carat-

teri o frasi. È fondamentale perciò muovere il cursore nella posizione desiderata prima di intervenire in tal senso; esistono comandi per spostare il cursore nelle posizioni adiacenti alla posizione attuale o muovere l'intera pagina video.

nuto di altri;

cp: (copy) è il comando di copia: permette di copiare un file in un altro, creando così una copia identica;

mv: (move) permette di cambiare il nome a un file (cioè di attribuirgli un nuovo path-name);

ln: (link) consente di generare un riferimento, cioè un ulteriore path-name, a un file già esistente; più semplicemente determina la presenza di un doppio nome (un "alias") per lo stesso file. Quindi il file-system di Unix prevede che lo stesso file possa essere identificato da più di un path-name. Questa operazione è detta "linking";

rm: (remove) è il comando di cancellazione dei file. Normalmente, se non richiesto esplicitamente, non consente di cancellare una direttrice. Se il file che si cancella ha qualche alias definito tramite il comando **ln**, l'effetto del comando **rm** è semplicemente quello di cancellare il nome indicato. Il contenuto del file sarà accessibile con gli altri nomi presenti nel file-system;

chmod: è il comando che consente di modificare le protezioni associate ai file.

Gli editor

Unix offre un'ampia scelta di programmi editor del tipo "general purpose", cioè adatti a ogni applicazione, dalla scrittura di un programma, a quella di un file di dati, di un testo, e così via. Tra questi editor, i più diffusi sono:

ed: è un "line editor" di tipo tradizionale; consente di operare sul file a livello di linea, dovendo sempre specificare la linea sulla quale si opera. È adatto per i terminali telescriventi, ma piuttosto scomodo se si dispone di un terminale video. Oggi, è utilizzato soprattutto per editare file in modalità background, cioè fornendo come input a **ed** un file di comandi al posto dell'input interattivo;

vi: è un "full screen editor": visualizza sul video una porzione del file che si sta correggendo. L'utente può spostarsi all'interno della pagina video con il cursore. La pagina stessa si muove all'interno dell'intero file (figura di pagina precedente). L'editor **vi** è molto apprezzato per la ricchezza di funzioni e per la velocità operativa.

I comandi per la gestione dei processi

Un corretto impiego dei processi, in modalità foreground o background, determina una maggiore capacità operativa. Unix offre alcuni comandi per la gestione dei processi:

ps: (process status) fornisce all'utente la lista dei processi presenti nel sistema e il loro stato;

kill: (kill) è il comando che permette di interrompere l'esecuzione di un processo. Spesso è utilizzato per interrompere processi in "loop" infinito;

tee: realizza la "derivazione da pipeline" qualora sia necessario conservare su un file i dati intermedi prodotti in una pipeline. Per esempio, supponiamo di voler eseguire il comando:

```
cat filea fileb: pr >/dev/lp &
```

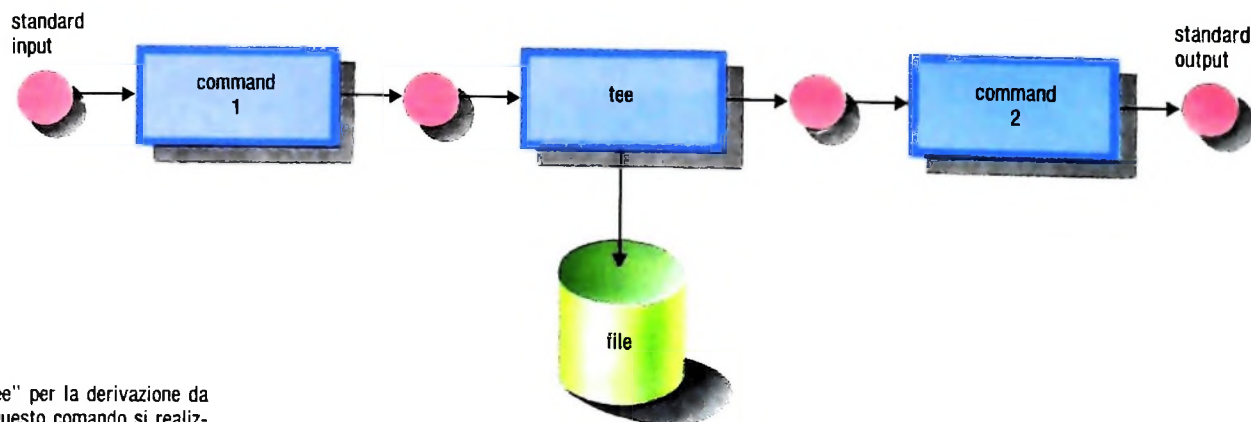
Il risultato del comando **cat** può essere conservato su di un file prima di essere rediretto nello standard input di "pr" così:

```
cat filea fileb: tee saved : pr > dev/lp &
```

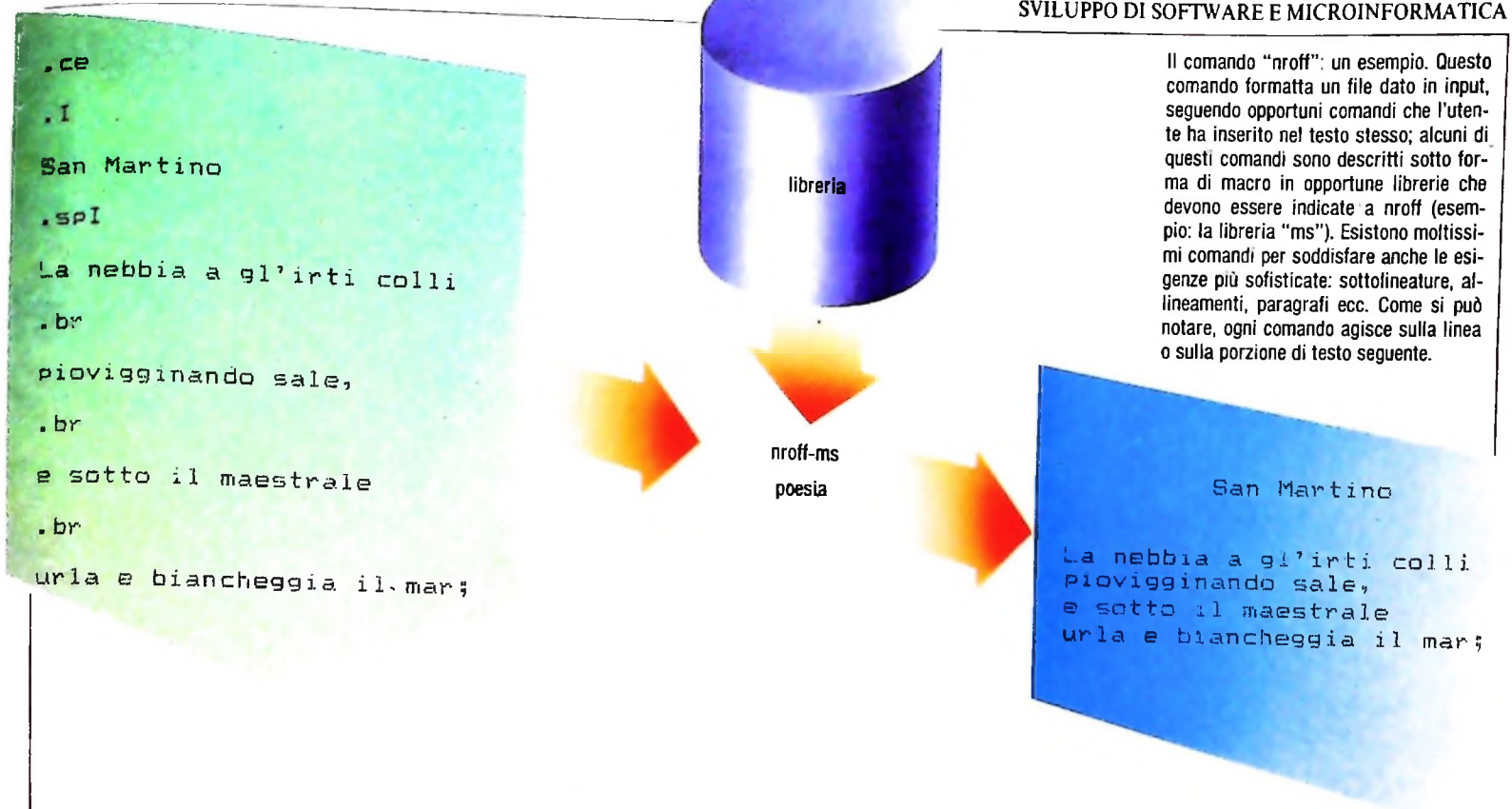
In altre parole, **tee** copia il proprio standard input sia nel proprio standard output sia nel file specificato (figura a destra).

I filtri

Di notevole importanza, i filtri sono quegli oggetti che elaborano lo standard input riproducendo, secondo la funzione



Il comando "tee" per la derivazione da pipeline. Con questo comando si realizza la "derivazione da pipeline" tutte le volte che si rende necessario conservare su un file i dati intermedi prodotti in una pipeline.



Il comando "nroff": un esempio. Questo comando formatta un file dato in input, seguendo opportuni comandi che l'utente ha inserito nel testo stesso; alcuni di questi comandi sono descritti sotto forma di macro in opportune librerie che devono essere indicate a nroff (esempio: la libreria "ms"). Esistono moltissimi comandi per soddisfare anche le esigenze più sofisticate: sottolineature, allineamenti, paragrafi ecc. Come si può notare, ogni comando agisce sulla linea o sulla porzione di testo seguente.

che devono svolgere, le informazioni generate sullo standard output. Ne esistono parecchi, in quanto sono spesso utilizzati nelle pipeline. Tra questi:

- wc:** (word count) fornisce il numero di linee, parole e caratteri che compongono l'input. È utile, per esempio, per statistiche su testi;
- sort:** ordina le linee dell'input, secondo il criterio indicato nelle sue possibili opzioni;
- prep:** spezza le linee del testo di input, suddividendole in modo che ogni linea del testo di output prodotto contenga una sola parola.

I comandi di stampa e visualizzazione

Per potere conoscere il contenuto dei file esistono degli appositi comandi:

- more:** è il visualizzatore per terminali video. Emette al terminale il contenuto dei file desiderati in forma paginata. L'utente ha a disposizione un insieme di comandi che vanno dalla selezione delle linee o pagine seguenti, alla ricerca di parole, all'attivazione di un editor sul file in esame;
- pr:** è un paginatore per terminali scriventi, come la stampante di sistema. Si utilizza per produrre i listati dei programmi, dei file di dati ecc.;
- lpr:** quando l'utente vuole accedere all'uso della stampante di sistema può far uso di questo comando, che stampa i file di tutti gli utenti, gestendo una apposita coda di attesa.

I comandi di "office automation"

Unix dispone di alcuni comandi che realizzano funzioni tradizionalmente considerate di "office automation":

- nroff:** è il formattatore base dei testi. Insieme ad altri comandi e ad alcune librerie apposite, permette all'utente di realizzare del "word and text processing". Questi strumenti hanno una loro complessità data dalla generalità degli output che si possono ottenere (figura di questa pagina, in alto);
- mail:** la posta elettronica è uno strumento di comunicazione tra gli utenti molto efficiente. Gli utenti possono, grazie a questo comando, scambiarsi dei messaggi e gestire un "archivio" dei messaggi ricevuti e/o inviati (figura di pagina seguente).

Gli strumenti di programmazione

Numerosi sono gli strumenti che semplificano le attività di sviluppo software. Facciamo due soli esempi:

- diff:** consente di conoscere quali sono le differenze tra il contenuto di due file generici. È molto utile quando, per esempio, un utente gestisce più versioni differenti dello stesso programma;
- grep:** un comune problema del programmatore è quello di conoscere quali file contengono alcune particolari stringhe: per esempio nomi di variabili. La questione è risolta da questo comando che segnala la presenza di specificati "pattern" all'interno di file.

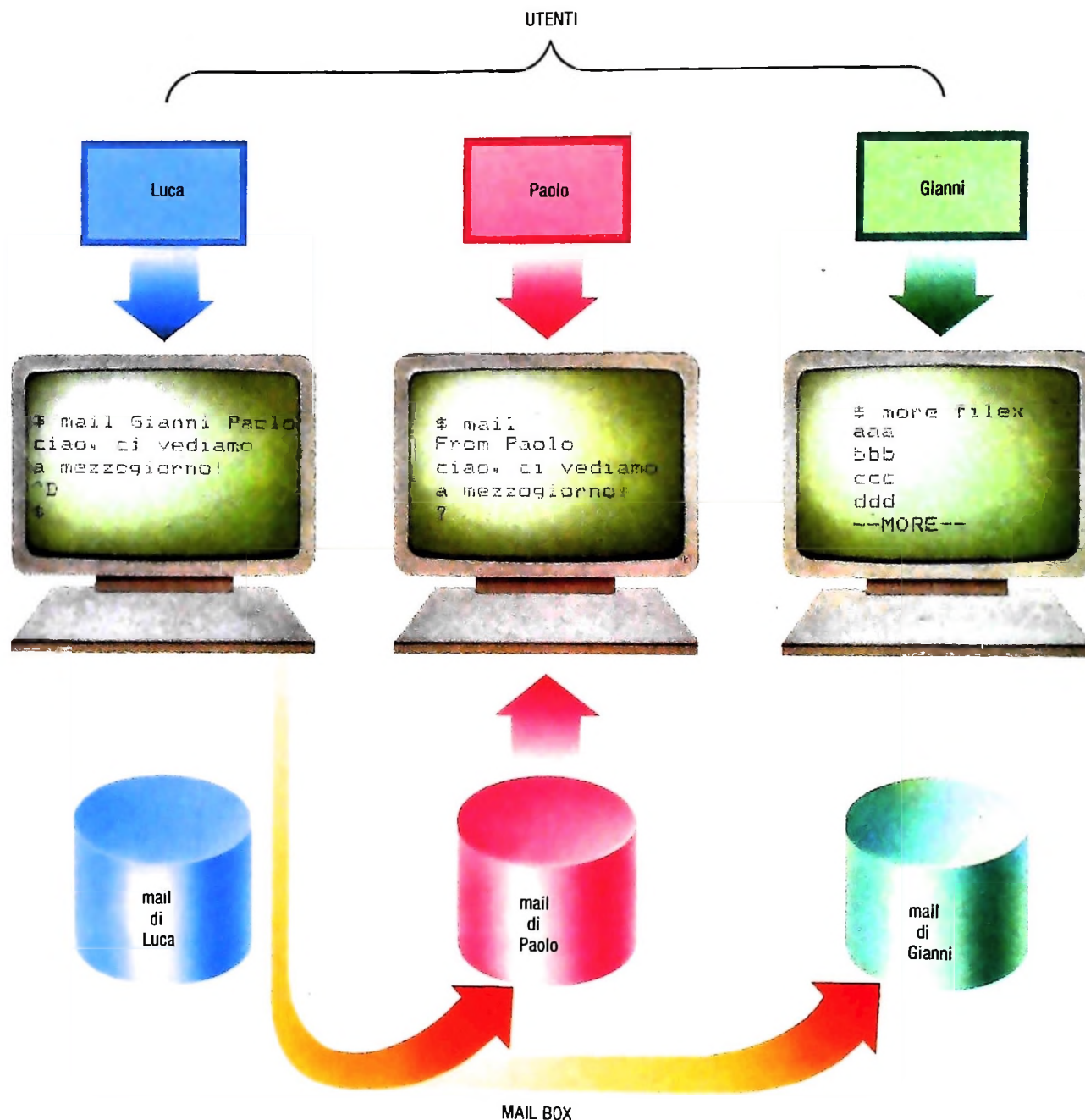
I linguaggi di programmazione disponibili

Unix dispone di numerosi linguaggi di programmazione, tra i quali: C, Pascal, Fortran, Basic, Cobol, Lisp. Il più usato è il C, in cui Unix stesso è implementato: è il linguaggio che meglio può sfruttare le potenzialità del sistema operativo. CC, il compilatore C, esegue quattro fasi distinte: una macroespansione, il controllo sintattico del sorgente, la traduzione in linguaggio assembler e l'ottimizzazio-

ne del codice prodotto.

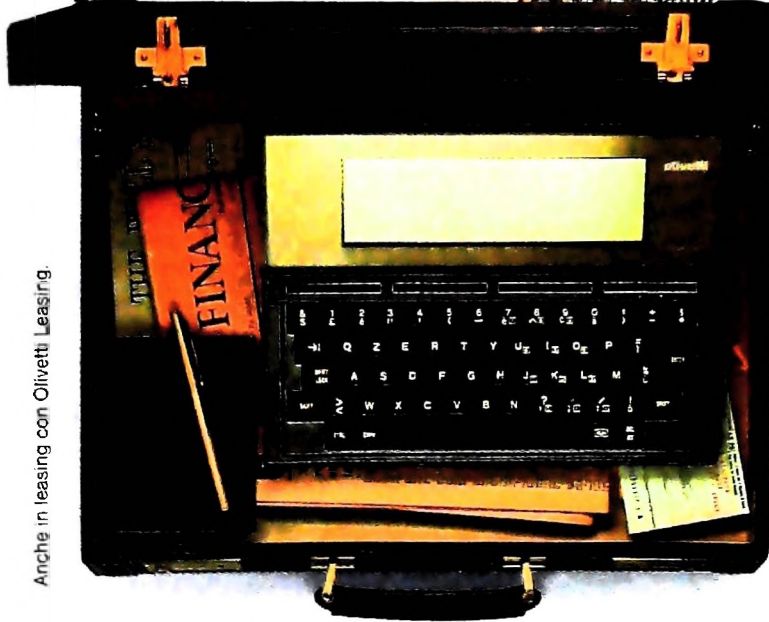
Si deve a Ken Thompson la prima versione del compilatore Pascal "pi" per Unix (1976).

Pi produce un codice che deve essere successivamente interpretato dall'interprete 'Px', per cui l'esecuzione è più lenta rispetto al codice prodotto dal compilatore C. È pure disponibile il compilatore f77 per il Fortran 77, linguaggio che si rifà ai concetti del Fortran classico ma ne amplia le possibilità: f77 e le sue librerie sono scritte in C.



Il comando "mail": un esempio. Questo comando consente scambi di messaggi tra gli utenti del sistema, ad ognuno dei quali è associato un file che funge da cassetta postale (mailbox). Nell'esempio, Luca invia da terminale contemporaneamente a Gianni e Paolo un testo libero: questo costituisce la "po-

sta" che è poi depositata nelle mailbox dei destinatari. Quando si desidera, la posta in arrivo può essere visualizzata, sempre col comando mail (come, in figura, fa l'utente Paolo.) Come si può notare, l'invio di un messaggio non interferisce con ciò che stanno facendo, in quel momento, i destinatari.



Anche in leasing con Olivetti Leasing.

PERSONAL COMPUTER OLIVETTI M10 L'UFFICIO DA VIAGGIO

Olivetti M10 vuol dire disporre del proprio ufficio in una ventiquattre. Perché M10 non solo produce, elabora, stampa e memorizza dati, testi e disegni, ma è anche capace di collegarsi via telefono per spedire o ricevere informazioni.

Qualunque professione sia la vostra, M10 è in grado, dovunque vi troviate, di offrirvi delle capacità di soluzione davvero molto grandi. M10: il più piccolo di una grande famiglia di personal.

olivetti

Per informazioni rivolgersi al negozio contrassegna via "Olivetti M10 Punto di Vendita" o inviare il coupon a: Olivetti, Divisione Personal Computer, Via Meravigli 12, 20123 Milano.

NOME COGNOME

VIA/N

CAP/CITTA

TELEFONO

— UN NUOVO MODO DI USARE LA BANCA. —

CONOSCIAMOCI MEGLIO



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

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

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

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

Veniteci a trovare, ci conosceremo meglio.

 **BANCO DI ROMA**
CONOSCIAMOCI MEGLIO.