

Cooperazione ad ambiente locale

di Luciano Macera

Siamo finalmente giunti alla puntata «centrale» della rubrica Multitasking. Da questo numero in poi, la nostra attenzione sarà focalizzata sui meccanismi di comunicazione a «scambio messaggi», propria della cooperazione ad ambiente locale, ma spesso presente anche in linguaggi di programmazione paralleli più propriamente ad ambiente globale

La netta distinzione

Nelle puntate precedenti abbiamo mostrato alcuni meccanismi di cooperazione-sincronizzazione basati sull'utilizzo (arbitrato) di strutture dati condivise. La condivisione di dati da parte di più processi paralleli è tipica della cooperazione ad ambiente globale. È proprio all'interno di questo che sono presenti le strutture comuni, e un certo numero di meccanismi messi a disposizione dal sistema permette di accedervi in maniera mutuamente esclusiva. Abbiamo così parlato di sospensione del multitasking, operazioni di Lock-UnLock, semafori, monitor: tutte con l'unico scopo di scongiurare disastrose collisioni tra processi paralleli.

A partire da questo numero, se volete, potete anche dimenticare tutto quanto narratovi (o quasi). Infatti nella cooperazione ad ambiente locale non esiste alcuna risorsa (o struttura) condivisa, ma tutto è regolato dai processi stessi. Processi, naturalmente, in grado di comunicare, ma sicuramente meno inclini a fare danni «in giro per il sistema». Ciò significa, tra l'altro, che non esistendo più strutture dati e risorse condivise, queste sono sempre e comunque inglobate all'interno di processi. Tanto per fare subito un esempio, se

due o più processi hanno bisogno di un buffer per effettuare operazioni di inserimento ed estrazione di oggetti, nel caso «ambiente globale» tale buffer sarebbe stata una risorsa condivisa il cui accesso viene regolato dalle solite P e V o da un monitor; nel caso «ambiente locale» è necessario creare un processo che al suo interno ha come struttura privata il buffer e tramite scambio messaggi con gli altri processi effettua inserimenti ed estrazioni di oggetti (figura 1).

Comunicazioni inter process

In figura 2 è mostrata schematicamente una coppia di processi in grado di comunicare tra loro. È il caso più semplice: abbiamo un processo A che produce dati da inviare a B che li utilizza. Sempre in figura 2 la freccia che collega i due processi è il canale di comunicazione tra questi. E non è, come potrebbe sembrare, una struttura condivisa dai due processi in quanto a livello di questi il canale non è visibile. È sì un oggetto condiviso, ma ad un livello più basso: al livello del nucleo del sistema operativo. Infatti tanto A quanto B non hanno visione del canale ma della comunicazione in una forma più astratta. A manda messaggi a B, B riceve mes-

saggi da A: che ci sia di mezzo un canale lo sappiamo noi e il sistema operativo, ma non i processi. Questo, naturalmente, quando la cooperazione è realmente ad ambiente locale: in realtà il più delle volte i linguaggi di programmazione parallela non sono così «puliti» fino in fondo e spesso lasciano visione all'interno dei processi anche di oggetti non meglio identificati ma comunque, almeno per certi versi, condivisi. Ma non scendiamo troppo nei dettagli subito: avremo modo di approfondire meglio l'argomento in seguito.

Dicevamo che il canale è un oggetto che permette la comunicazione tra processi ma non è direttamente visibile da questi. Un po' come il Program Counter nella programmazione in Assembler (quando questo non è allocato in un normale registro di macchina, ndr) o i vari puntatori alle aree dati nei linguaggi di programmazione ad alto livello.

Il canale è dunque direttamente (nonché esclusivamente) utilizzato dal nucleo del sistema operativo per effettuare la comunicazione.

Ciò che avviene all'interno dei processi (come in figura 2) è che il processo mittente ad un certo punto esegue una operazione di «send» per spedire un messaggio (ad esempio un dato) al processo B. Questo, per ricevere l'informazione in arrivo da A, analogamente eseguirà la sua operazione di comunicazione che nel caso del processo destinatario sarà una «receive». La «send» avrà tra i suoi parametri l'oggetto da spedire, la «receive» una variabile dello stesso tipo nella quale riceverà l'oggetto trasmesso. In pratica a trasmissione avvenuta l'effetto finale sarà quello di assegnare alla variabile indicata nella funzione «receive» del processo destinatario il valore dell'oggetto trasmesso dal mittente e indicato nella funzione «send».

Il canale, conseguentemente, eredita il tipo dell'oggetto trasmesso che come abbiamo detto è anche uguale a quello della variabile targa del processo destinatario.

Ma per definire un canale non basta indicare il suo tipo ma occorre in qualche modo definire anche mittente e destinatario della comunicazione. Esistono fondamentalmente due strade: canali con nomi espliciti dei processi (figura

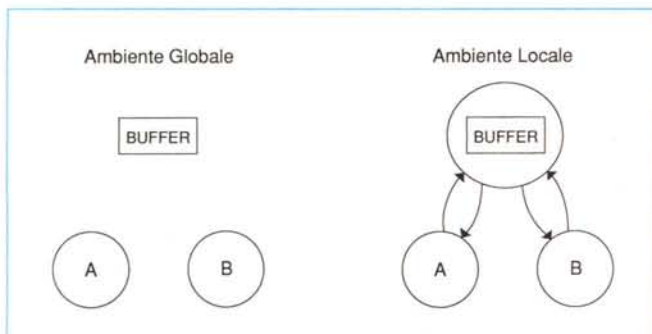


Figura 1 - Nella cooperazione ad ambiente globale le risorse condivise vengono arbitrate da meccanismi di mutua esclusione, in quella ad ambiente locale le risorse sono inglobate nei processi.

3a) e canali con porte (figura 3b). Ricordando che tale definizione non spetta né al processo mittente né al processo destinatario di una comunicazione ma semplicemente al sistema operativo che deve avere un «quadro chiaro» di tutta la situazione, è al momento della compilazione dei vari processi che sono definiti implicitamente anche i vari canali.

Nel caso di canali con nomi espliciti dei moduli il canale è definito dalla tripla ordinata:

(Mittente, Destinatario, Tipo)

e viene dedotta dal sistema in base alle operazioni di «send» presenti nei processi mittenti e alle operazioni di «receive» nei processi destinatari che contengono come primo parametro il nome del processo partner. Se ad esempio nel processo A troviamo:

send(B, messaggio)

nel processo B:

receive(A, variabile)
e tanto «messaggio» quanto «variabile» sono dello stesso tipo T, il canale è definito dalla tripla:

(A,B,T)

Nel caso di canali con porte il canale è definito dalla coppia ordinata:

(PortaMittente, PortaDestinatario)

Le porte sono oggetti privati dei processi e rappresentano delle interfacce logiche tra i processi e i canali di comunicazione. Hanno lo stesso tipo di messaggio da trasmettere e quindi di canale che interfacciano.

Nelle operazioni di «send» presenti nei processi mittenti e nelle operazioni di «receive» nei processi destinatari si fa esplicito riferimento non ai processi partner ma alla propria porta di ingresso o di uscita precedentemente definita.

Ciò che è ora necessario è «linkare», prima dell'utilizzo, una porta con un processo, sia nel caso del mittente (che lin-

ka la sua porta al nome del processo destinatario) che nel caso del destinatario (che linka la sua porta al nome del processo mittente). Se ad esempio nel processo A troviamo:

out port PA: T
var messaggio: T
bind(PA, B)
send(PA, messaggio)

nel processo B:

in port PB: T
var variabile: T
bind(PB, A)
receive(PB, variabile)

il canale è definito dalla coppia:

(PA, PB)

e non è più presente il tipo in quanto già definito nella dichiarazione delle porte.

Comunicazioni sincrone e asincrone

C'è un particolare che abbiamo volutamente sorvolato fino ad ora: la sincro-

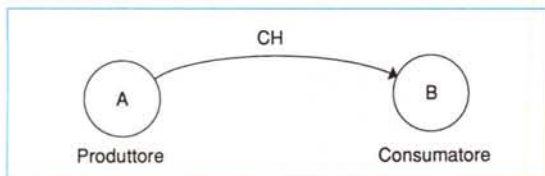


Figura 2 - Nella cooperazione ad ambiente locale le comunicazioni inter process avvengono esclusivamente attraverso messaggi trasmessi, ricevuti e canali di comunicazione.

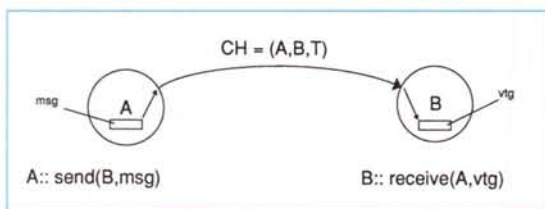


Figura 3a - Canali con nomi espliciti dei processi. In questo caso il canale è definito dalla tripla ordinata (Mittente, Destinatario, TipoDelMessaggio).

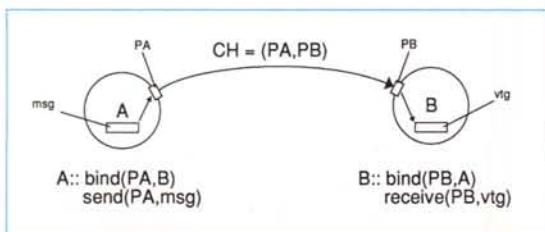


Figura 3b - Canali con porte. Qui il canale è identificato dalla coppia di porte (PA,PB) rigorosamente dello stesso tipo e connesse mediante l'esecuzione di «bind» corrispondenti nei due processi.

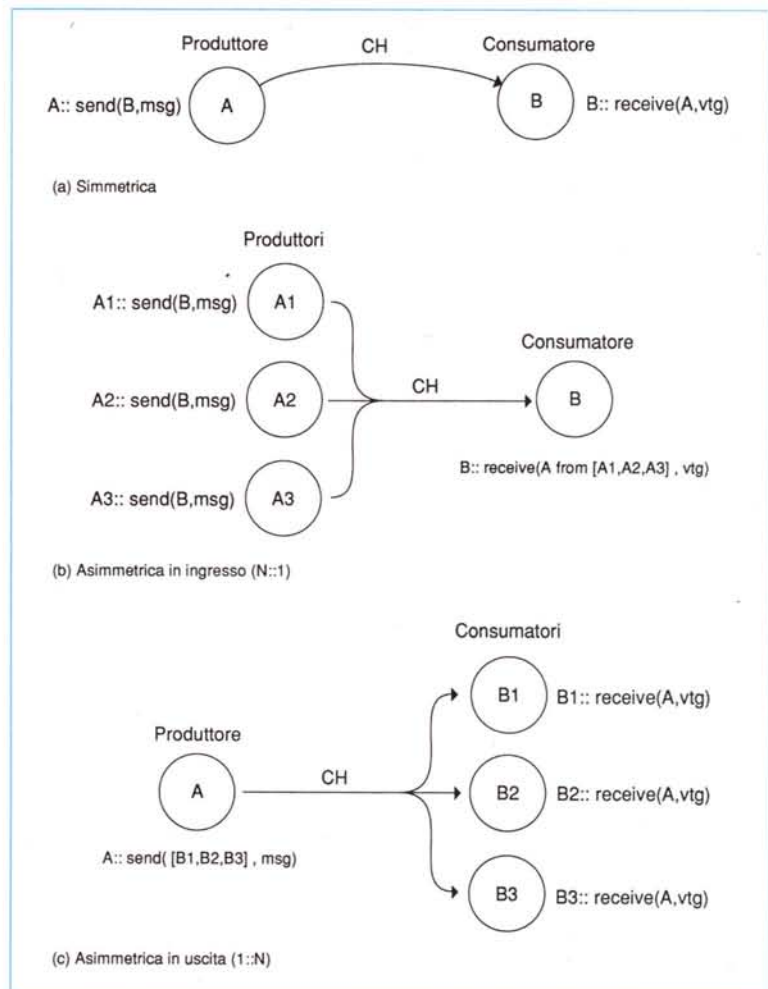


Figura 4 - Forme di comunicazione.

na o asincronia delle comunicazioni. Cosa succede, per dirla in breve, se, quando il mittente effettua una «send», il destinatario non effettua la corrispondente «receive» perché momentaneamente indaffarato in altri lavori? Il mittente deposita il messaggio e continua oppure attende pazientemente il destinatario per consegnare personalmente il messaggio? Esistono in genere tutt'e due le possibilità: e, naturalmente, non esiste la soluzione migliore, ma possono essere utili l'una o l'altra possibilità a seconda dei casi. È chiaro, inoltre, che la comunicazione asincrona è possibile solo associando al canale un'area di memoria dove immagazzinare i messaggi spediti e non ancora letti.

Discorso del tutto analogo per le operazioni di «receive». Cosa deve fare il destinatario se un messaggio richiesto non è stato ancora spedito? Resta lì impalato ad aspettare, oppure continua per la sua strada senza il messaggio richiesto? Come nel caso precedente può essere utile disporre di entrambe le soluzioni a seconda dei casi. Nei prossimi numeri, quando cominceremo a mostrare alcuni programmi multitask vedremo come utilizzare a seconda dei casi le varie chance disponibili.

Forme di comunicazione

In figura 4 sono mostrate le forme di comunicazione base. La prima, la più semplice, prevede semplicemente che vi sia nella comunicazione un solo mittente e un solo destinatario (comunicazione simmetrica). È il caso che abbiamo già utilizzato negli esempi precedenti in cui un processo invia e un processo riceve attraverso un canale. Naturalmente è possibile instaurare una nuova comunicazione anche dal processo destinatario verso il mittente (invertendo i ruoli) attraverso un nuovo canale diverso dal primo. Infatti, posto che il tipo dei dati che i due processi si scambiano vicendevolmente è lo stesso, i due canali sono diversi essendo differenti il mittente e il destinatario nei due casi. Il primo canale sarà ad esempio identificato dalla terna:

(A, B, T)

il secondo dalla terna:

(B, A, T)

Le forme di comunicazione asimmetriche (rappresentate in figura 4b e 4c) consentono rispettivamente l'esistenza di più mittenti o più destinatari sul medesimo canale. La comunicazione asimmetrica in uscita (figura 4c) è detta anche comunicazione per diffusione in quanto il medesimo messaggio è inviato simultaneamente a più destinatari.

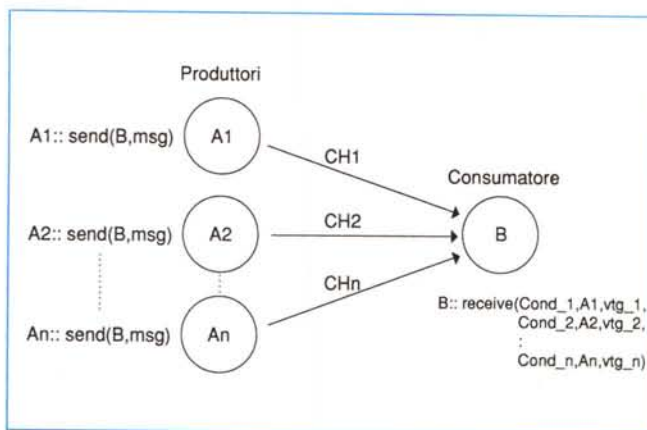


Figura 5
Controllo del
nondeterminismo.

Nel primo caso della comunicazione asimmetrica in ingresso, si tratta di messaggi diversi (provenienti da processi diversi) che si accodano sul canale e che verranno letti in istanti successivi dal processo destinatario. Tanto i mittenti di figura 4b quanto destinatari di figura 4c non sono a conoscenza dell'asimmetria del canale: per loro la comunicazione è sempre e comunque simmetrica (il loro partner nella comunicazione è unico). Viceversa il destinatario di una comunicazione asimmetrica in ingresso e il mittente di una comunicazione asimmetrica in uscita sono coscienti della asimmetria in quanto il primo dovrà esplicitamente indicare da quali processi (più d'uno) riceve, il secondo a quali processi spedire.

Il nondeterminismo

Se fino ad oggi pensavate che gli attuali calcolatori fossero delle macchine deterministiche, avete una visione rigidamente sequenziale dei calcolatori.

Oggi, fermo restando la determinicità di un singolo processo (che ad ogni sequenza di input fornisce sempre un'unica sequenza di output), le cose stanno ben diversamente nelle macchine multitasking. Infatti a livello di processi nessuna ipotesi è fatta sull'avanzamento parallelo di questi, né sull'indeterminatezza di alcune situazioni che possono verificarsi. L'esempio tipico è mostrato proprio in figura 4b: i processi A1, A2, A3 sono assolutamente indipendenti; utilizzano (senza saperlo) lo stesso canale verso B il quale riceve sequenzialmente i messaggi dai tre processi mittenti. Se nello stesso istante logico più processi inviano messaggi a B non è definito a priori in quale sequenza questi saranno recapitati al destinatario. Né B potrà fare nulla per forzare una sua preferenza o priorità.

E se in alcuni casi possiamo procedere incuranti del problema, in altri potrebbe essere necessario pilotare in qualche modo la scelta del messaggio da ricevere per primo.

Naturalmente in questo caso non

avremo più un unico canale asimmetrico ma tanti canali simmetrici quanti sono i mittenti (figura 5). All'interno del destinatario la receive avrà come parametri tante triple quanti sono i mittenti.

Ogni tripla sarà formata da una variabile targa (nella quale, eventualmente, ricevere il messaggio), il nome del mittente e una variabile di condizionamento (ad esempio un'espressione logica) che maschera o meno la lettura da quel canale. Così se abbiamo 5 canali e altrettanti processi che spediscono e vogliamo ricevere solo dai primi due processi sarà sufficiente far valere TRUE i primi due parametri booleani e FALSE i rimanenti tre.

Niente paura

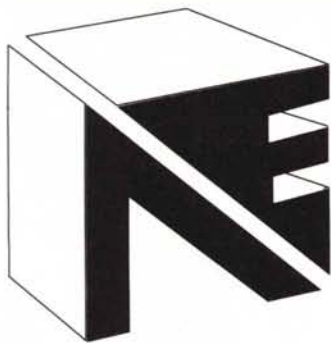
Certo, per chi ha utilizzato per proprie risorse di calcolo sempre in maniera sequenziale questo linguaggio potrà sembrare anche un tantino ostico, ma per fortuna una volta presa la mano tutto torna ad essere semplice e magicamente affascinante.

Tra gli utenti dei piccoli sistemi i più fortunati sono gli utenti Amiga che dispongono di un vero e proprio computer multitask con tanto di processi, messaggi, porte, risorse, anche se non sempre gestiti o gestibili nel più pulito dei modi.

Gli utenti MS-DOS per giocare «al multitasking» hanno invece una possibilità diversa, molto più interessante, anche se poco proponibile dal punto di vista economico: acquistare una scheda per il loro PC dotata di almeno un transputer e due mega di ram e utilizzare su questa il linguaggio OCCAM.

Così come faremo noi a partire dal prossimo numero: metteremo un po' il naso nel multitask che si circonda per spulciare insieme le cose più interessanti dal punto di vista didattico e applicativo. Inutile ricordarvi che sono, come sempre, ben accette tutte le collaborazioni dei lettori che perverranno in redazione. Per il momento buon lavoro e appuntamento al prossimo mese.

MS



NEWEL® srl

computers ed accessori 20155 MILANO via Mac Mahon, 75

NEGOZIO tel.0 2 / 3 2 3 4 9 2
UFFICI tel.0 2 / 3 2 7 0 2 2 6
FAX 24h tel.0 2 / 3 3 0 0 0 3 5
UFFICIO **SPEDIZIONI**
 tel.0 2 / 3 3 0 0 0 3 6

VENDITA ANCHER PER CORRISPONDENZA IN TUTTA ITALIA
EVASIONE ORDINI NELLE 24 ORE SUCCESSIVE ALL'ORDINE

HARDCARD 2091 (SOLO CONTROLLER) L. 330.000
HARDCARD 2091 + 45 MB HARDISK "QUICKLOAD"
 L. 875.000

Hard-disk Commodore nuovo veloce controller, espandibile a 2 MB, il tutto su CARD, semplicissima installazione, autoboot, il tutto ad un prezzo eccezionale.

AMIGAVISION (NOVITÀ)

Finalmente disponibile anche in Italia il straordinario Pacchetto che fino ad oggi corredeva soltanto l'Amiga 3000, questo eccezionale programma per la programmazione multimediale, grafica, animazione, video, musica e molto più. Regala al tuo Amiga questo fantastico programma!!!

L. 149.000

NUOVE SCHEDE ACCELERATRICI "GVP A-3001" ANCORA PIU' VELOCI!!!!

Nuovo ora 33 o 50 Mhz. Accelera il tuo Amiga A2000 più di dieci volte, rendendolo notevolmente più veloce del nuovo A3000, corredato di 4 MB di ram a 32 Bit, e coprocessore 68882, completo di controller per Hard-disk.

NELLA VERSIONE 33MHZ. L.3.390.000

NELLA VERSIONE 50MHZ. L.4.990.000

KIT. SUPPLEMENTARE PER ALTRI 4 MB A 32 BIT. L. 899.000

C'E BULK E BULK NOI TI OFFRIAMO DISCHETTI DI QUALITÀ 3 1/2 DS-DD CERTIFICATI UNO AD UNO.

SONY CONF. DA 50 PZ. L. 900 CAD.
 SONY CONF. DA 100 PZ. L.800 CAD.
 SONY CONF. DA 200 PZ. L.700 CAD.

TUTTI I PREZZI IVA COMPRESA

VIDEOMASTER

Il più straordinario Genlock professionale per tutti gli amiga, Qualità Broadcast, S-VHS (luminanza cromaticità separati) già in console effetti video.

NOVITÀ MONDIALE

HARD DISK INTERNO PER A500

Probabilmente il più piccolo HARD DISK da 20 MB al mondo!!! - Si inserisce all'interno del AMIGA 500 - Alta velocità 20 MB drive - Formattato per amiga DOS - Autoboot sotto Kickstart 1.3 - Autoparking - Compatibile con Adspeed accelerator - Semplice installazione
 TUTTO QUESTO AD UN PREZZO ANCORA

MANUALE IN ITALIANO **AMIGA PC AT EMULATOR** **ORA ANCHE PER A2000**
 ATONCEPC 286 EMULATOR AT AMIGA 500 MULTITASKING
Lire 390.000

+ ESPANSIONI DI MEMORIA PER AMIGA 500, 1000 E 2000

Costruite con i migliori materiali, le nuove espansioni di memoria dell'ultima generazione usano i nuovissimi chip da 1 mbit che sono notevolmente più veloci, autoconfiguranti, slim line, e con 1 anno di garanzia!

- 512k per amiga 500 L. 85.000
- 512k per amiga 500 + clock L. 110.000
- 1,5 mb per amiga 500 + clock L. 240.000
- 4 mb per amiga 500 +clock L. 590.000
- 2 mb per amiga 1000 L. 450.000
- 2 mb espandibile a 8 per amiga 2000 + clock L. 390.000

UTILITY DISK DEL VALORE DI L. 50.000 IN OMAGGIO

MODEM RS-232 STANDARD HAYES COMPATIBILI:

- ESTERNO SMART 300/1200 (V21-V22) L. 179.000
- ESTERNO SMART 300/1200/VIDEOTEL (V21-V22-V23) L. 249.000
- ESTERNO SMART 300/1200/2400 (V21-V22-V22 BIS) L. 249.000
- ESTERNO SMART 300/1200/2400 con correzione errore MNP-5 L. 369.000
- ESTERNO SMART 300/1200/2400/VIDEOTEL (V21-V22-V23-V22BIS) L. 399.000-

DISPONIBILI ANCHE TUTTE LE VERSIONI IN SCHEDA INTERNA!!!

DAVA INTER/STARNI PER TUTTI GLI AMIGA ALTA QUALITÀ

GARANZIA 12 MESI

- DRIVE INTERNO PER AMIGA 500 L. 149.000
- DRIVE ESTERNO PER AMIGA 500/1000 PASS. + DISCONNECT L. 149.000
- DRIVE INTERNO PER AMIGA 2000 (COMPLETO) L. 139.000
- DRIVE ESTERNO PER AMIGA 500/1000/2000 DA 5"1/4 L. 249.000

UTILITY DISK DEL VALORE DI L. 50.000 IN OMAGGIO

DA OGGI NEWEL È ANCHE ATARI "ST"

- NUOVO ATATI STE 1040 + OMAGGIO "GARANZIA ITALIANA" L. 990.000
- FLOPPY DISK DRIVE 720K ESTERNO L. 179.000
- HAND SCANNER 105 mm. + SOFTWARE TOUCH UBI [NOVITA'] L. 449.000
- MOUSE DI RICAMBIO "SUPERMOUSE" L. 69.000
- MOUSE OTTICO ALTA PRECISIONE [NOVITA'] L. 99.000
- HARD DISK "QUICKLOAD 20 MB" Superveloce [NOVITA'] L. 699.000
- HARD DISK "QUICKLOAD 40 MB" Superveloce [NOVITA'] L. 899.000
- HARD DISK "QUICKLOAD 110 MB" Superveloce [NOVITA'] L.1.399.000
- BLITZ "SYNCRO" Nuovissimo copiatore Hardware per Atari ST L. 49.000
- AT-ONCE Nuovo emulatore Pc per Atari L. 390.000
- TRACK-BALL-ST comodissimo trackball in alternativa al mouse! L. 99.000
- Prosound Digitalizzatore Audio Stereofonico L. 110.000
- JOYSTICK Dedicato per ATARI ST L. 10.000

richiedi il nostro nuovo catalogo gratuitamente specificando il computer posseduto!

Tutte le ultime novità software, originali, importazione diretta!!!
 Disponibile in magazzino l'intera gamma professionale Amiga 3000

TUTTI I NOSTRI PRODOTTI SONO COPERTI DA GARANZIA DI 12 MESI
TUTTI I PREZZI SONO IVA 19% COMPRESA
COMMODORE AMIGA 500 E 2000 AI PREZZI PIU' BASSI D'ITALIA.

SUPER SYNCRO EXPRESS V. 3.0

Nuova versione del più potente copiatore Hardware, con nuovo CHIP castum, vi permette di effettuare copie di sicurezza ad uso strettamente personale della maggior parte del software protetto, opzione Quickcopy in meno di 1 minuto, semplice installazione, funziona su Amiga 500 & 1000. L. 89.000

A-2320 FLICKER FIXER

Elimina il fastidioso sfarfallio del Amiga in altissima risoluzione, indispensabile per chi lavora in grafica, è indispensabile un monitor multisync o VGA. L.430.000

SUPER SYNCRO EXPRESS V. 3.0

Nuova versione del più potente copiatore Hardware, con nuovo CHIP castum, vi permette di effettuare copie di sicurezza ad uso strettamente personale della maggior parte del software protetto, opzione Quickcopy in meno di 1 minuto, semplice installazione, funziona su Amiga 500 & 1000. L. 89.000

INTERFACCIA 4 JOYSTICK

Permette di collegare contemporaneamente 4 joystick all'amiga e quindi di giocare in 4 contemporaneamente a giochi tipo CALCIO, PALLAVOLO ecc. DISPONIBILE !!! L. 29.000

ACTION REPLAY 2

(disponibile anche per AMIGA 2000)

la prima cartuccia rivoluzionaria multifunzioni per amiga 500/1000 con opzioni di freeze: permette di sprotteggere la maggior parte dei programmi in commercio (consentendoti di creare giochi di sicurezza per uso personale, inoltre permette di creare giochi trainer, (vite infinite ecc.), permette di bloccare un gioco in qualsiasi momento dal medesimo posto, salva una qualsiasi videata (disegno, testo) su disco, consentendoti una facile hardcopy anche su stampante, funzione moviola (rallenta programmi e giochi), potente virus-detector, sprinteeditor, oltre che ad un monitor straordinario per il linguaggio macchina, questo e molto, molto di più, ti aspetta in amiga action replay!!! il tutto ad un prezzo eccezionale! versione originale con manuale in italiano L. 169.000

AutoCAD

VERSIONI DALLA 2.5 ALLA 10.0
CORSO DI PROGRAMMAZIONE IN AutoLISP
INTRODUZIONE ALLA VERSIONE 11.0

**PC
MASTER**

CORSI COMPLETI IN AUTOISTRUZIONE

Oggi esiste un metodo sicuro e immediato per apprendere tutte le tecniche di disegno assistito da computer:

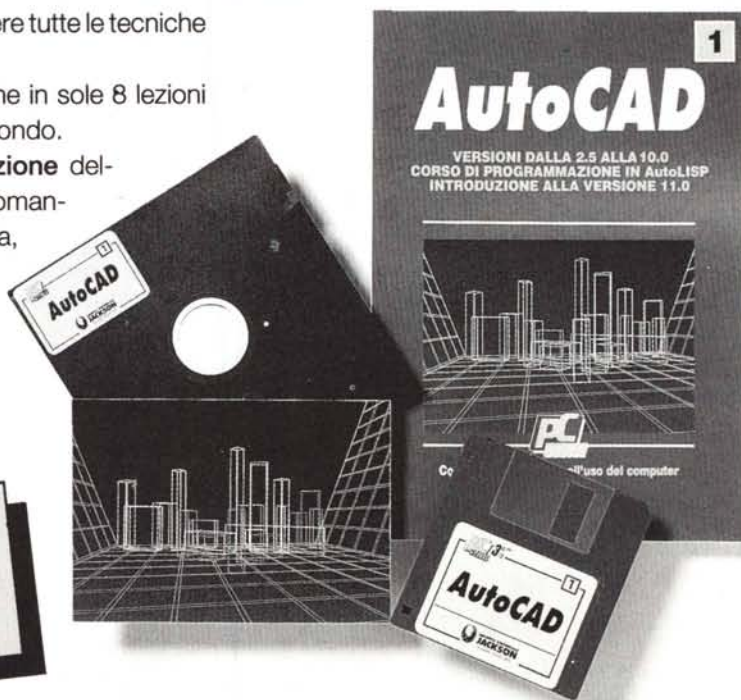
è il **corso in autoistruzione PC Master AutoCAD**, che in sole 8 lezioni interattive insegna il programma CAD più diffuso nel mondo.

Attraverso **8 fascicoli** ed un **programma di simulazione** dell'ambiente AutoCAD, il lettore impara l'uso corretto dei comandi operando immediatamente all'interno del programma, guidato e assistito passo passo dal proprio computer.

Ogni lezione presenta una **sezione principianti** nella quale vengono trattati esclusivamente i comandi e le funzioni di AutoCAD, e una **sezione esperti** nella quale vengono approfonditi i concetti relativi ad AutoLISP, l'avanzatissimo linguaggio di programmazione di AutoCAD.

**PCMaster
AutoCAD
8 lezioni:
8 testi + 8 dischi
in fascicoli
quindicinali**

In edicola
nelle due
versioni
5 1/4" e 3 1/2"



GRUPPO EDITORIALE
JACKSON