a cura di Andrea de Prisco

Il controllo della concorrenza

di Anna Pugliese

seconda puntata

Il problema del Controllo della Concorrenza, presentato sul precedente numero di MC, nasce con l'uso del parallelismo fra diverse attività in esecuzione su un sistema che utilizzano risorse comuni alle attività stesse.
Two phases locking e Timestamping sono i nomi dei due più classici meccanismi di controllo della concorrenza realmente utilizzati dai computer attualmente in commercio. Il primo di essi verrà presentato in questo numero della rubrica, il secondo assieme agli ultimi sviluppi della ricerca in materia, saranno oggetto di una successiva ed ultima parte dedicata a questo tema.
L'articolo comprende la presentazione di alcuni problemi strettamente legati al controllo della concorrenza, fra i quali il famoso deadlock

Una precisa visione del problema

È doveroso richiamare alcuni punti trattati il mese scorso. Primo fra tutti, l'individuazione degli attori: Transazioni e Risorse.

La transazione è un costrutto di programmazione mediante il quale è possibile racchiudere una generica sequenza di operazioni, ognuna delle quali può o meno agire su di un dato condiviso, in una specie di unica, grande operazione.

Un importante aspetto delle transazioni è il loro punto di commit. Per chiarire questo concetto occorre tener conto del fatto che le transazioni sono uno strumento mediante il quale è possibile trattare sia il problema del controllo della concorrenza che quello della tolleranza ai guasti dei sistemi di elaborazione. Dal punto di vista della concorrenza, una transazione è una quantità indivisibile d'elaborazione, l'effetto della sua esecuzione deve coincidere con quello che si sarebbe ottenuto eseguendo la transazione in assenza di concorrenza. Dal punto di vista dell'affidabilità, una transazione è una quantità indivisibile d'elaborazione il cui effetto, anche in presenza di guasti, dev'essere o nullo oppure identico a quello di una corretta esecuzione dell'intera transazione. Guardando le cose in maniera integrata, l'esecuzione di una transazione corrisponderà all'esecuzione di tutte le operazioni, di cui essa è composta, non sui dati effettivi ma su copie dei dati stessi. In seguito all'esecuzione dell'ultima operazione verrà poi inserita un'operazione di commit, il cui scopo è quello di convalidare gli effetti della transazione. Tale convalida avviene solo nel caso in cui non si sono verificati errori e non si sono create inconsistenze sui dati per effetto di un'errata serializzazione, in tal caso la transazione viene COMMITTATA, altrimenti viene ABORTITA. Sul sistema per la gestione della base di dati, in un generico istante, saranno in esecuzione un certo numero di transazioni concorrenti molte delle quali si troveranno ad agire su archivi comuni. Tali archivi saranno allora considerati come risorse condivise dalle transazioni.

Considerando il succedersi degli eventi dal punto di vista di una precisa risorsa, è facile comprendere che le modifiche apportate al suo valore non possono essere considerate assolute, ma relative alla transazione che ha generato la modifica. Le operazioni sulle risorse saranno allora considerate come coppie (op, tid), dove op (OPeration) specifica il tipo di operazione invocata sulla risorsa e tid (Transaction IDentifier) l'identificatore associato alla transazione richiedente.

Ora, ogni operazione su un dato condiviso è di per sé un'operazione critica; essa va eseguita con particolari accorgimenti quali ad esempio il locking del dato condiviso durante l'operazione. Il locking di un dato è un'operazione di privatizzazione temporanea della struttura contenente il dato stesso. Essa viene implementata associando ad ogni dato condiviso una chiave (to lock = chiudere a chiave) dal cui valore è possibile capire se il dato è stato o meno privatizzato temporaneamente da qualche transazione. Tutto ciò ha lo scopo di impedire l'esecuzione contemporanea di operazioni diverse sullo stesso oggetto. Il lettore interessato ad una più approfondita trattazione di questo argomento è rimandato a questa stessa rubrica pubblicata sul numero 52 di MC.

Data per scontata l'adozione di opportuni meccanismi che garantiscano che in un certo istante su un certo oggetto esiste al più una operazione in esecuzione, lo scopo da prefiggersi è quello di permettere che su un certo oggetto in un certo istante sia possibile mantenere in esecuzione tutte le transazioni che si vuole, vale a dire eseguire in sequenza operazioni appartenenti a transazioni diverse. Tutto questo, a condizione di mantenere l'integrità dell'oggetto stesso.

L'esecuzione seriale

Dato un certo numero di transazioni, il più semplice metodo per eseguirle senza problemi è quello di utilizzare uno scheduling seriale. Con riferimento alla figura 1, basterà implementare un processo scheduler (ordinatore) al quale tutte le transazioni dovranno essere indirizzate per essere da esso stesso eseguite, se-

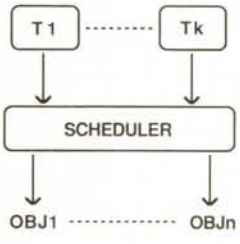
rialmente, mediante l'invocazione delle operazioni, specificate in ogni transazione, all'oggetto corrispondente. È evidente che tale processo costituisce un collo di bottiglia per il sistema, e che il parallelismo nell'esecuzione delle transazioni è, in tal caso, perso.

Il locking a due fasi

Il locking a due fasi, anche conosciuto con la sigla 2PL che sta per Two Phases Locking, è un meccanismo per il controllo della concorrenza molto particolare, oltre ad essere il più diffuso fra quelli esistenti. La sua peculiarità risiede nel fatto che esso è un meccanismo statico, vale a dire che la sua funzionalità è dovuta al modo in cui la transazione è strutturata.

Esso è basato sul requisito, cui ogni transazione deve soddisfare, di suddividere la transazione in due fasi: la growing phase (fase crescente) e la shrinking phase (fase decrescente). Durante la prima fase, la transazione può richiedere tutti i lock di cui necessita per svolgere le sue operazioni ogni volta che un'operazione su un nuovo oggetto viene generata.

Figura 1



Tali lock devono essere trattenuti fino a quando la transazione raggiunge il suo punto di commit, a questo punto ha inizio la seconda fase della transazione consistente nel rilascio di tutti i lock precedentemente richiesti.

Il parallelismo esplicitato con l'uso di questo meccanismo di controllo della concorrenza dipende dalle collisioni fra le richieste di lock sugli oggetti avanzate da parte delle varie transazioni. Se su un certo oggetto è in esecuzione un'operazione di una transazione, tale oggetto non potrà essere acceduto da nessun'altra transazione fino al completamento della transazione che ne possiede il lock.

Se ne ricava che non esiste parallelismo sugli oggetti. Al più saranno eseguibili in parallelo transazioni operanti su insiemi di oggetti disgiunti.

```
procedure PRENOTATION ( flight, day );
begin
N := FREE ( flight, day );
if N > 0 then
                begin
                FREE ( flight, day ) := N-1;
                Result := "done"
                end
          else
                Result := "not done"
end;
```

Figura 2.
La procedura di prenotazione vista il mese scorso, che attivata due volte concorrentemente causa inconsistenze sui dati.

2PL con prerilascio

Una variante del metodo 2PL è la cosiddetta versione con prerilascio. Essa è basata sulla possibilità di rilasciare i lock posseduti dalla transazione prima ancora di giungere al punto di commit, non appena l'oggetto del quale si intende prerilasciare il lock non è più necessario alla transazione stessa, ma a condizione che in seguito al rilascio di un qualsiasi lock non vengano richiesti lock su nessun altro oggetto. Quest'ultima condizione è necessaria per garantire la strutturazione a due fasi della transazione, senza la quale l'esecuzione potrebbe generare inconsistenze nei dati.

Riguardiamo l'esempio

Entrambe le versioni del locking a due fasi sono pensate per operare con transazioni che coinvolgono un qualunque numero di operazioni su un qualunque numero di oggetti. Nondimeno può essere utile vederne l'applicazione su un caso molto semplice qual è quello presentato la volta scorsa. La procedura è riportata in figura 2, essa ha lo scopo di prenotare un posto sul volo specificato dal parametro flight per il giorno indicato da day, FREE(i,j) è l'elemento della matrice bidimensionale, che costituisce la risorsa condivisa, indicante il numero dei posti ancora disponibili sul volo i per il giorno j.

In figura 3 sono riportare le due procedure scritte in accordo ai due meccanismi di controllo della concorrenza precedentemente illustrati.

Gli aborti in cascata

Potrebbe, a prima vista, sembrare ovvia la convenienza ad utilizzare la versione del 2PL con prerilascio rispetto a quella base. Tuttavia essa presenta un inconveniente molto sottile, ma che in ambienti in cui l'uso del parallelismo fra le transazioni è massiccio, potrebbe degradare la performance del sistema a livelli inaccettabili. Si consideri a tal proposito e nel caso in cui il meccanismo di controllo della concorrenza utilizzato è il 2PL con prerilascio, una transazione T1 che dopo aver prerilasciato un certo numero di lock su oggetti da essa modificati nel corso della sua esecuzione, venga abortita a causa, si supponga, di un errore. Ciò renderà necessario abortire anche eventuali altre transazioni che avevano acquisito il lock prerilasciato da T1, essendo la correttezza della loro esecuzione subordinata al buon esito della transazione abortita. Se si considera la possibilità che queste nuove transazioni da abortire, prima ancora che vengano effettivamente abortite, abbiano modificato e prerilasciato oggetti condivisi, si comprende facilmente come questo possa dar vita ad una serie di aborti, detti appunto in cascata, dagli effetti certamente disastrosi dal punto di vista dell'efficienza.

```
        procedure PRENOTATION ( flight, day );
        ---------------------------------------
begin                                  ¦¦ begin
LOCK (FREE(flight,day));               ¦¦ LOCK (FREE(flight,day));
N := FREE(flight,day );                ¦¦ N := FREE(flight,day);
if N > 0                               ¦¦ if N > 0
then                                   ¦¦ then
  begin                                ¦¦   begin
  FREE(flight,day) := N-1;             ¦¦   FREE(flight,day) := N-1;
  Result := "done"                     ¦¦   UNLOCK (FREE(flight,day));
  end                                  ¦¦   Result := "done"
else                                   ¦¦   end
  Result := "not done";                ¦¦ else
UNLOCK (FREE(flight,day))              ¦¦   Result := "not done"
end;                                   ¦¦ end;
-------------------------------------------------------------------
            (a)                                    (b)
-------------------------------------------------------------------
```

Figura 3 - La procedura illustrata in figura 2 scritta in accordo al metodo Two Phases Locking, rispettivamente in versione base (fig. 3a) ed in versione con prerilascio (fig. 3b).

```
Ta::   lock(X)          ;t1 ; Tb::   lock(W)
       (opX,ta)         ;t2 ;        (opW,tb)
       lock(Y)          ;t3 ;        lock(Z)
       (opY,ta)         ;t4 ;        (opZ,tb)
       lock(Z))         ;t5 ;        lock(Y)
       unlock(X)        ;t6 ;        unlock(W)
       (opZ,ta)         ;t7 ;        (opY,tb)
       unlock(Z)        ;t8 ;        unlock(Y)
       (opY,ta)         ;t9 ;        (opZ,tb)
       unlock(Y)        ;t10;        unlock(Z)
       COMMIT           ;t11;        COMMIT
```

Figura 4.
Due transazioni agenti sugli oggetti X, Y, Z, e W. L'espressione (opX,ta) designa una qualsiasi operazione invocata dalla transazione ta sull'oggetto X. Le due transazioni sono strutturate in accordo al meccanismo di controllo della concorrenza 2PL con prerilascio. Si noti la strutturazione a due fasi delle transazioni: da t1 a t5 GROWING PHASE e da t6 a t11 SHRINKING PHASE.

Deadlock

Consideriamo ora le due transazioni descritte in figura 4. Esse sono strutturate in accordo al meccanismo di locking a due fasi in versione con prerilascio, ma quanto detto nel seguito vale anche nel caso della versione 2PL con ritenzione fino al commit.

Per non appesantire troppo la descrizione, si assuma che la durata di tutte le operazioni sia fissa e che nel generico istante Ti (1 <= i <= 11), come indicato in figura, vengano eseguite in parallelismo reale le i-esime operazioni di Ta e Tb.

Fino all'istante t4 le prime due operazioni di Ta e le prime due operazioni di Tb saranno eseguite senza problemi, essendo operazioni su oggetti distinti.

Nell'istante t5 la transazione Ta richiederà il lock sull'oggetto Z prima di poter procedere nell'esecuzione della sua terza operazione. Tale lock non potrà essere ottenuto da Ta in quanto già assegnato, fin dall'istante t3, alla transazione Tb che lo tratterrà fino a t10, a motivo della necessità di riutilizzare l'oggetto Z nell'istante t9. La transazione Ta verrà dunque sospesa in attesa che tale lock venga rilasciato. Purtroppo ciò non avverrà mai in quanto la transazione Tb, nello stesso istante verrà a trovarsi in un'aloga situazione in attesa del lock sull'oggetto Y posseduto dalla transazione sospesa Ta.

La situazione che si è venuta a verificare nell'esempio descritto, prende il significativo nome di DEADLOCK (chiusura della morte). In effetti le due transazioni resteranno per sempre bloccate in attesa dei rispettivi lock, a meno che non si intervenga dall'esterno per sbloccare la situazione.

L'esempio presentato rispecchia uno dei modi più semplici in cui il deadlock può verificarsi, essendo solo due le transazioni coinvolte. In realtà non esiste un limite al numero di transazioni che possono rimanere coinvolte in simili situazioni di stallo. Il fenomeno del deadlock è comune a tutti i meccanismi di controllo della concorrenza, paradossalmente anche a quelli non basati sul lock esplicito degli oggetti. Per difendersi da questo fenomeno esistono due possibilità: la prevenzione e la rilevazione del deadlock.

La prevenzione del deadlock consiste nell'utilizzare meccanismi capaci di impedire il verificarsi del fenomeno. Un simile meccanismo, anche se in forma del tutto implicita, è utilizzato dal metodo di controllo della concorrenza basato sui timestamp. Per ciò che riguarda il 2PL invece, i tentativi di utilizzare prevenzione del deadlock hanno spesso avuto esiti insoddisfacenti. A mo' di

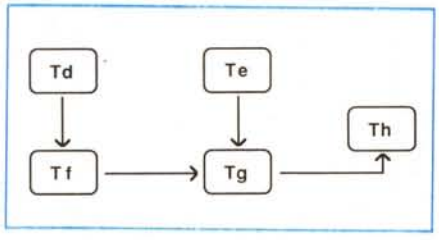


Figura 5 - Un esempio di wait-for graph.

cronaca tuttavia, si noti che un semplicissimo modo per farlo consiste nell'imporre il requisito di richiedere tutti i lock necessari all'inizio della transazione; se sono stati ottenuti tutti allora si procede, diversamente si rilasciano anche i pochi lock acquisiti e, dopo un certo intervallo di tempo, si ritenta l'acquisizione.

Più diffusa è invece l'utilizzazione di meccanismi di rilevazione del deadlock. Fra di essi, il più elegante consiste nella costruzione dei cosiddetti «WAIT-FOR GRAPHES» vale a dire grafi di precedenza (letteralmente: attesa per). In tali grafi, i nodi corrispondono alle transazioni, mentre gli archi diretti da una transazione in attesa ad un'altra in possesso del lock per cui la prima attende, esprimono proprio la condizione dei wait-for.

Un semplice modo per implementarli, è quello di affidarne l'aggiornamento alle stesse transazioni, nel momento in cui esse, eseguendo la primitiva di lock su un dato oggetto trovano l'oggetto stesso occupato da un'altra transazione. In tal caso la transazione che ha trovato l'oggetto occupato, prima di sospendersi, aggiornerà il grafo di precedenza, che potrebbe anche essere vuoto, aggiungendo ad esso un nodo che porta il suo nome, un altro nodo che porta il nome della transazione in possesso del lock cercato, sempre che tali nodi non esistano già, ed un arco che va dal suo nodo a quello dell'altra. Infine, la transazione stessa effettuerà un controllo sul grafo così ottenuto. Se in tale grafo esiste un ciclo, il deadlock è stato rilevato. Basterà a questo punto forzare l'aborto di una delle transazioni coinvolte nel ciclo, in accordo a qualche predefinita politica.

Con riferimento alla figura 5, si supponga che quello in essa riportato sia l'attuale grafo di precedenza. Sul sistema saranno in esecuzione concorrente un numero di transazioni maggiore o ugule al numero dei nodi presenti sul grafo. Se a questo punto, la transazione Th, in possesso di un lock voluto anche da Tg, si sospende in attesa di un lock posseduto da Te, essa dovrà aggiornare il grafo con un arco direzionato da Th a Te, la qual cosa provocherà l'insorgere di un ciclo nel grafo stesso, a testimonianza del verificarsi del deadlock. Si noti che il deadlock non coinvolge solo le transazioni nel ciclo, ma anche Td e Tf che attendono per Tg.

Per quanto riguarda il deadlock illustrato in figura 4, la semplicità della sua dinamica è testimoniata dal grafo di precedenza, illustrato in figura 6, che si sarebbe venuto a formare in tal caso.

Un simile meccanismo di rilevazione del deadlock è certamente assai elegante, come già accennato sopra, ma sicuramente dispendioso. L'aggiornamento del grafo da parte delle transazioni, ma ancor più il test di ciclicità potrebbero far trascorrere più tempo di quanto sarebbe necessario per avere il lock dispo-

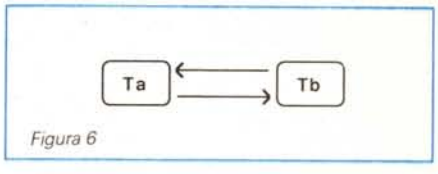


Figura 6

nibile. L'alternativa più diffusa, ai grafi di precedenza, consiste nell'uso di timeout: se una transazione attende per un tempo maggiore di una soglia definita a priori, decide di abortire se stessa per sbloccare un'eventuale situazione di deadlock. L'uso di timeout è diffuso un po' in tutte le aree di produzione del software, ma il problema di definire in maniera ottima la soglia di tempo di un timeout, oramai diventato storico nel campo dell'informatica, non è mai stato risolto brillantemente. Che un timeout alla fine, sia più o meno efficiente di un wait-for graph, è solo questione di fortuna. MC

Il Computer è servito!

postalbit

VENDITA PER CORRISPONDENZA
Terminal Centro e Sud d'Italia
Tel. 06/5424303

PERSONAL COMPUTER 386
Personal Computer MsDos, OS/2 e Xenix compatibile.
Microprocessore 80386 a 16 Mhz. Velocita' 20 Mhz.
Coprocessore matematico opzionale. Tastiera 101 tasti.
Memoria RAM 2 Mb on board. Alimentatore. Cabinet Tower con spie di funzionamento, tasti di selezione e chiave.
Controller per Floppy disk drive e Hard Disk. Scheda video.
Manuale operativo. Garanzia 12 mesi.
Configurazioni disponibili: Un Floppy Disk Drive 1,2 Mb.
Un Floppy disk drive 1,2 Mb e hard disk 20/40/70 Mb.

Configurazioni e prezzi

386/1	Un Fd 1,2Mb	Lire	5.020.000
386/2	Un Fd 1,2Mb Hard Disk 20Mb	Lire	5.437.000
386/3	Un Fd 1,2Mb Hard Disk 40Mb	Lire	6.773.000
386/4	Un Fd 1,2Mb Hard Disk 70Mb	Lire	8.609.000

Parti per Computer Kit

Cabinet XT/AT completo di alimentatore	Lire	193.000
Case Tower completo di alimentatore	Lire	821.000
Mother Board Turbo Xt 0K	Lire	130.000
Mother Board AT 13 Mhz 0K	Lire	450.000
Mother Board 386 20Mhz 2Mb	Lire	6.450.000
Floppy Disk Controller Xt	Lire	75.000
Hard Disk Controller Xt	Lire	197.000
Hard/Floppy Disk Controller At	Lire	209.000
Scheda video grafica colore con prister	Lire	80.000
Scheda video monocromatica con prister	Lire	80.000
Scheda video EGA	Lire	247.000
Scheda video PGA	Lire	300.000
Scheda video VGA	Lire	432.000
Scheda seriale	Lire	35.000
Chip abilitazione seconda porta seriale	Lire	25.000
Scheda seriale,giochi,mouse,clock,prister	Lire	75.000
Scheda joystick	Lire	30.000
Scheda Clock real time	Lire	55.000
Scheda modem 300/1200 baud	Lire	150.000

PERSONAL COMPUTER 8088
Personal Computer Ms Dos. Microprocessore Intel 8088 Turbo a 4,7/8 Mhz. Coprocessore matematico opzionale.Tastiera 101 tasti. Memoria Ram 256K espandibile a 1024 K su scheda madre. Alimentatore 150W. Cabinet flip top metallico.
Pannello di controllo esterno con tasto reset e tasto turbo.
Chiave di sicurezza. Spie di funzionamento. Controller per due floppy disk drive. Scheda video.
Manuale operativo . Garanzia 12 mesi
Opzioni : Coprocessore matematico. Espansioni di memoria.
Scheda video EGA o PGA. Monitor vari. Stampanti.

Configurazioni e prezzi

8088/1	Un FDD 360K	Lire	767.000
8088/2	Due FDD 360K	Lire	897.000
8088/3	Un FDD 360K e Hard Disk 20Mb	Lire	1.283.000
8088/4	Un FDD 720K	Lire	788.000
8088/5	Due FDD 720K	Lire	933.000
8088/6	Un FDD 720K e Hard Disk 20Mb	Lire	1.303.000
8088/7	Un FDD 360K e un FDD 720K	Lire	813.000

Memorie di massa

Floppy disk drive 360K	Lire	120.000
Floppy disk drive 720K	Lire	175.000
completo di adattatore		
Floppy disk drive 1,2Mb	Lire	212.000
Hard Disk Drive 20Mb 3"1/2	Lire	408.000
Hard Disk Drive 40Mb 40Ms	Lire	737.000
Hard Disk Drive 70Mb 28Ms	Lire	1.615.000
Back Up 20Mb con cassetta	Lire	1.165.000
Cassetta per Back Up 20Mb	Lire	80.000

Add In / Add On

Deviatore 2 Persoonal 1 Stampante	Lire	80.000
Mouse Meccanico Microsoft Comp.le	Lire	98.000
Mouse Logitec Compatibile	Lire	81.000
Joystick per PC	Lire	27.000
Gruppo di continuita' 250W	Lire	263.000
Gruppo di continuita' 500W	Lire	550.000
Coprocessore matematico 8087/5	Lire	340.000
Coprocessore matematico 80287/8	Lire	675.000
Cavo Centronics per stampante	Lire	70.000

PERSONAL COMPUTER 80286
Personal Computer MsDos/OS2/Xenix compatibile.
Microprocessore Intel 80286 a 6/8/10 Mhz. Velocita' 13 Mhz.
Coprocessore matematico opzionale. Tastiera 101 tasti.
Memoria RAM 512K espandibile 1 Mb su scheda madre. Cabinet Flip Top metallico. Pannello di controllo esterno con tasto di reset e turbo. Spie di funzionamento. Chiave di sicurezza
Controller per Floppy Disk Drive e Hard Disk. Scheda video
Manuale operativo. Garanzia 12 mesi.
Configurazioni disponibili: Un Floppy Disk Drive 1,2 Mb.
1 Floppy Disk Drive da 1,2 Mb e Hard disk da 20/40/70 Mb.
Opzioni: Coprocessore matematico. Espansioni di memoria.
Schede video EGA e PGA. Monitor vari. Stampanti.

Configurazioni e prezzi

286/1	Un FDD 1,2 Mb	Lire	1.552.000
286/2	Un FDD 1,2 Mb HD 20 Mb	Lire	1.968.000
286/3	Un FDD 1,2 Mb HD 40 Mb	Lire	2.825.000
286/4	Un FDD 1,2 Mb HD 70 Mb	Lire	3.686.000

Personal Computer Philips

PP0
Personal Computer IBM Compatibile
Processore Intel 8088 a 4,7/8 Mhz
RAM 768K On board. Scheda video Monocromatica e Grafica Colori. Scheda Seriale e Parallela.
Real Time clock. Controller HD su mother board
Tastiera. Manuale d'uso. MS DOS e GW BASIC.

PP1	Con 2 FDD 720K	Lire	1.365.000
PP2	Con 1 FDD 720K e HD 20 Mb	Lire	1.829.000

Personal Computer Travel

Personal Computer IBM Compatibile formato valigetta.
Microprocessori 8088 o 80286. Schermo LCD con scheda video grafico colore e alta risoluzione grafica EGA. Floppy disk drive da 360K o 720K o 1,2Mb. Dischi rigidi da 20 e 40 Mb.
Tastiera. Controllo di contrasto e tasto reset.
Uscita per monitor e tastiera esterni e stampante.

Configurazioni e prezzi

TRV1	XT Turbo 256K 2 FDD 360K	Lire	1.892.000
TRV2	XT Turbo 256K 1 FDD 360K HD20Mb	Lire	2.408.000
TRV3	XT Turbo 256K 2 FDD 720K	Lire	1.992.000
TRV4	XT Turbo 256K 1 FDD 720K HD20Mb	Lire	2.408.000
TRV5	AT 13Mhz 512K 1 FDD 1,2M HD20Mb	Lire	2.872.000
TRV6	AT 13Mhz 512K 1 FDD 1,2M HD40Mb	Lire	3.209.000

Monitor

M4	Data Engineering TTL/CGA Bianco Carta	Lire	260.000
M5	Data Engineering Colori Media Ris.ne	Lire	500.000
M6	Data Engineering Colori E.G.A.	Lire	710.000
M7	Data Engineering Multisync Colori	Lire	950.000
M8	Philips Basculante FOsfori Verdi/Ambra	Lire	360.000
M9	Philips Colori E.G.A.	Lire	630.000
M10	Samsung Basculante Monocromatico	Lire	146.000

Stampanti OKI Microline 9/18 aghi

ML182P	Stampante ad aghi 80 colonne 120 cps bidirezionale grafica NLQ int.parall	Lire	550.000
ML182S	Come sopra con interfaccia seriale	Lire	609.000
ML192P	Stampante ad aghi 80 colonne 200 cps bidirezionale grafica NLQ int.parell con inseritore frontale fogli singoli	Lire	720.000
ML192S	Come sopra con interfaccia seriale	Lire	858.000
ML193P	Stampante ad aghi 136 colonne 200 cps bidirezionale grafica NLQ int.parall con inseritore frontale fogli singoli	Lire	880.000
ML193S	Come sopra con interfaccia seriale	Lire	1.000.000
ASM001	Inseritore automatico fogli singoli 80 colonne per 192/193	Lire	430.000
ASM002	Inseritore automatico fogli singoli 136 colonne per 193	Lire	580.000
ASM003	Kit trattore/schermo acustico per modelli 182/192	Lire	175.000
ASM004	Cartuccia nastro nero 182/192/193	Lire	25.000
ML292	Stampante 18 aghi 80 col. 240 cps bidirezionale grafica NLQ stampe in nero o a colori interfaccia parallela o seriale	Lire	890.000
ML293	Come sopra 136 colonne	Lire	1.150.000
ML294	Come sopra 400 cps	Lire	1.850.000
ASM005	Inseritore automatico fogli singoli 80 colonne per 292/293	Lire	450.000
ASM006	Inseritore automatico fogli singoli 136 colonne per 293	Lire	580.000
ASM007	Kit trattore/schermo acustico 292	Lire	175.000
ASM008	Cartuccia nastro nero 292	Lire	30.000
ASM009	Cartuccia nastro colore 292	Lire	35.000
ASM010	Cartuccia nastro nero 293/294	Lire	40.000
ASM011	Cartuccia nastro colore 293/294	Lire	45.000

Stampanti Oki Microline 24 Aghi

ML393	Stampante 24 aghi 136 col 360 cps bidirezionale grafica NLQ interfaccia parallela e seriale	Lire	2.000.000
ML393C	Come sopra ma a colori	Lire	2.700.000
ML390P	Stampante 24 aghi 80 col 270 cps bidirezionale grafica NLQ int.par.	Lire	1.140.000
ML390S	Come sopra ma interfaccia seriale	Lire	1.260.000
ML391P	Come 390P ma a 136 colonne	Lire	1.420.000
ML391S	Come sopra con interfaccia seriale	Lire	1.550.000
ASM012	Inseritore singolo automatico 393	Lire	650.000
ASM013	Inseritore doppio automatico 393	Lire	1.000.000
ASM014	Inseritore singolo automatico 391	Lire	620.000
ASM015	Inseritore doppio automatico 391	Lire	1.000.000
ASM016	Inseritore singolo automatico 80col per 390/391	Lire	820.000
ASM017	Come sopra doppio	Lire	1.130.000
ASM018	Cartuccia Font 393/390/391	Lire	120.000
ASM019	Kit trattore opzionale 393	Lire	190.000
ASM020	Kit trattore opzionale 390	Lire	120.000
ASM021	Kit trattore opzionale 391	Lire	140.000
ASM022	Cartuccia nastro nero 393	Lire	55.000
ASM023	Cartuccia nastro color 393	Lire	65.000
ASM024	Cartuccia nastro nero 390/391	Lire	27.000

Stampanti OKI Microline Laser

LS61	Stampante laser 6 pagine minuto RAM 512K Stampe alta qualita' Risoluzione 300x300 Interfaccia parallela o seriale Emulazioni HPLjet II, Epson, IBM, Diablo Qume, NEC	Lire	3.800.000
LS62	Come sopra con RAM 1,5 Mb	Lire	4.180.000
LS63	Come sopra con RAM 2,5 Mb	Lire	4.750.000
LS121	Come LS61 12 pagine minuto	Lire	5.330.000
LS121	Come sopra con RAM 2Mb	Lire	6.200.000
ASM25	Memoria RAM 1,5 Mb per LS6	Lire	750.000
ASM26	Memoria RAM 2,5 Mb per LS6	Lire	1.500.000
ASM27	Vassoio alimentazione carta 500 fogli per LS6	Lire	1.150.000
ASM28	Cartuccia font per LS6	Lire	290.000
ASM29	Cartuccia font per LS12	Lire	450.000
ASM30	Cartuccia generazione barcode LS12	Lire	1.200.000
ASM31	Cartuccia emulazione plotter LS12	Lire	1.000.000
ASM32	Espansione memoria RAM 1,5 Mb LS12	Lire	1.500.000
ASM33	Gruppo pulizia con tamburo e filtro ottico per LS6	Lire	350.000
ASM34	Come sopra senza tamburo e filtro	Lire	180.000
ASM35	Toner per LS6	Lire	45.000
ASM36	Gruppo pulizia, stampa, olio fusione per LS12	Lire	650.000
ASM37	Gruppo fusione per LS12	Lire	350.000
ASM38	Toner per LS12	Lire	55.000

Diskettes e Moduli Continui

DM1	10 Diskettes Postalbit 5"1/4 DSDD	Lire	9.000
DM2	10 Diskettes Postalbit 3"1/2 DSDD	Lire	20.000
DM3	10 Diskettes Postalbit 5"1/5 HD	Lire	40.000
DM90	1000 Fogli 80 colonne bianco	Lire	16.500
DM91	1000 Fogli 136 colonne bianco	Lire	23.000
DM92	1000 Fogli 80 colonne lettura fac.	Lire	16.500
DM93	1000 Fogli 136 colonne lettura fac.	Lire	23.000
DM94	500 Fogli Uso Bollo	Lire	19.000
DM95	200 Buste in modulo continuo	Lire	26.500
DM96	500 Ricevute bancarie a norme ABI	Lire	15.000

Accessori EDP

A1	Sottostampante plexiglass 80 col	Lire	40.000
A2	Sottostampante plexiglass 136 col	Lire	60.000
A3	Sottostampante tubolare 80 col	Lire	25.000
A4	Sottostampante tubolare 136 col	Lire	30.000
A5	Portadischetti 3" 1/2 10 posizioni	Lire	2.500
A6	Portadischetti 3" 1/2 40 posizioni	Lire	12.500
A7	Portadischetti 3" 1/2 80 posizioni	Lire	21.000
A8	Portadischetti 5" 1/4 10 posizioni	Lire	2.500
A9	Portadischetti 5" 1/4 50 posizioni	Lire	12.500
A10	Portadischetti 5" 1/4 90 posizioni	Lire	19.000
A11	Base in gomma antistatica per mouse	Lire	18.000
A12	Base in gomma antistatica tastiera	Lire	30.000
A13	Base in gomma antistatica computer	Lire	13.000
A14	Piastra girevole monitor e computer	Lire	35.000
A15	Schermo filtrante per monitor 12"	Lire	75.000
A16	Schermo filtrante per monitor 14"	Lire	83.000
A17	Kit di Pulizia per Computer	Lire	19.500
A18	Kit di Pulizia per Drive 5" 1/4	Lire	12.000
A19	Kit di pulizia per Drive 3" 1/2	Lire	17.000
A20	Gas purissimo spray	Lire	6.500
A21	Copritastiera PC	Lire	16.000
A22	Copritastiera PC 101 tasti	Lire	18.000
A23	Copritastiera PC AT	Lire	18.000
A24	Copritastiera Olivetti M19/M28	Lire	18.000
A25	Copritastiera Olivetti M24	Lire	18.500
A26	Copricomputer PVC per PCXT	Lire	11.000
A27	Copricomputer PVC per PCAT	Lire	15.000
A28	Copricomputer PVC per Olivetti M19	Lire	15.000
A29	Copricomputer PVC per Olivetti M24	Lire	15.000
A30	Copricomputer PVC per Olivetti M28	Lire	15.000

TUTTI I PREZZI SONO IVA ESCLUSA

Conservate questo listino - Resterà valido fino alla prossima uscita