

# ADPnetwork: concludendo...

di Andrea de Prisco

*Siamo finalmente giunti all'ultimo appuntamento, su queste pagine, riguardante la nostra rete fault tolerant per computer Amiga. In quest'articolo faremo una specie di riassunto generale, non tanto dal punto di vista tecnico-implementativo, ma da quello dell'utente collegato (giustamente) in rete. Differentemente, poi, da quanto annunciato lo scorso mese, non siamo ancora in grado di fornirvi particolari sulla commercializzazione del prodotto (con l'estate di mezzo, proprio non è il caso...) ma possiamo sicuramente anticiparvi che ADPnetwork avrà un prezzo molto, molto «politico». Resta comunque valido l'invito, rivolto a tutti gli interessati, a prendere direttamente contatto col sottoscritto qui in redazione per qualsiasi tipo di chiarimento riguardante la rete*

## Presente e futuro

Come più volte riferito nei precedenti articoli, ADPnetwork utilizza attualmente l'interfaccia seriale a 31250 baud quale link fisico tra i vari nodi. A causa di questa limitazione hardware imposta dall'architettura Amiga non siamo riusciti a spingerci oltre tale velocità, ma stiamo studiando già da un pezzo una scheda hardware che, oltre a spingere la rete ad una velocità più di trenta volte superiore, eseguirà autonomamente il forward di tutti i messaggi in transito per i nodi. Contiamo, tra l'altro, di essere pronti con la versione hw addirittura al prossimo SMAU dove invitiamo tutti gli interessati a venirci a trovare nel nostro stand.

Attualmente con una rete software a 31250 baud possiamo sbizzarrirci quando le operazioni da compiere non sono particolarmente «lunghe». Quindi per applicazioni dove sono richieste «tante» operazioni «brevi» non esistono particolari problemi. Diciamo che accedere ad un file presente su un hd disponibile su una macchina remota è quasi come accedere ad un file presente su una unità a floppy disk locale.

In pratica è sempre meglio di correre da una macchina all'altra col floppino in mano per effettuare i trasferimenti di file.

Con la futura (ma non troppo) versione hardware non ci saranno più limitazioni di alcun tipo. Nel vero senso della parola potremo, da ogni macchina in rete, contare su tutti i device disponibili su tutte le macchine, come fossero nostri device senza notare alcun degrado delle prestazioni dovute alla presenza della rete. Certo se più macchine accedono alla stessa unità questa dovrà servire più richieste contemporanea-

mente con conseguenti ritardi sull'esecuzione, ma questo banalmente già succede anche su un singolo Amiga quando più processi in esecuzione su quella macchina accedono allo stesso device. In altre parole, montando ADPnetwork su un certo numero di macchine ciò che otteniamo è un nuovo ambiente multiuser, multiprocessor, sul quale grazie al net handler di Ciuchini e Suatoni possiamo addirittura contare su un file system distribuito, completamente compatibile con tutto il software già esistente. Ma questo è l'argomento del prossimo paragrafo...

## Compatibilità

In un certo senso dobbiamo ringraziare ancora una volta il sistema operativo di Amiga e la sua possibilità di montare facilmente nuovi device. Chi segue Amiga dai tempi del 1000 e del software di sistema versione 1.1 ricorderà che allora gli unici device disponibili erano i floppy disk (df0: ...df3:) la ram disk (RAM:), le porte seriale e parallela (SER:, PAR:) e il driver di stampa (PRT:). Non esisteva ancora la mountlist e per aggiungere un nuovo device bisognava smanettare nelle liste di sistema con appositi programmi ad hoc.

Poi con le release 1.2 e 1.3 sono nati tutta una serie di nuovi device (PIPE:, AUX:, VD0:, SPEAK:, ecc. ecc.) che spingevano l'utilizzo del file system anche verso nuovi orizzonti.

Quando poi i vari handler sono davvero AmigaDOS-compatibili le funzioni offerte rasentano l'incredibile. Pensate ad esempio a CrossDos che monta due nuovi device «A:» e «B:» per una perfetta compatibilità col file system MS-DOS. Caricate sul vostro Amiga un qualunque WP e, inserito un dischetto MS-

DOS nel drive «DF0:», chiedete di leggere o salvare un file nel device «A:» di CrossDos: il WP eseguirà senza nemmeno pensare (ammesso che sia possibile) di leggere o salvare file in formato MS-DOS. Tutti i comandi che AmigaDOS manderà al device «A:» saranno trasformati dall'handler di questo in comandi «compatibili» col formato del disco inserito nel drive che nulla ha a che vedere, appunto, con AmigaDOS.

Così un qualsiasi programma nato per girare e utilizzare file in formato AmigaDOS grazie all'aggiunta di un handler è in grado di leggere e scrivere file in un formato completamente diverso.

Grazie a questo meccanismo è nato il device «net:» che identifica, appunto, la nostra rete. Questo è visto dal sistema operativo di Amiga come una normalissima unità a dischi sulla quale sono possibili praticamente tutte le operazioni normalmente possibili su dischi. E se da shell diamo un bel

```
dir net:
oppure la sequenza di comandi:
cd net:
```

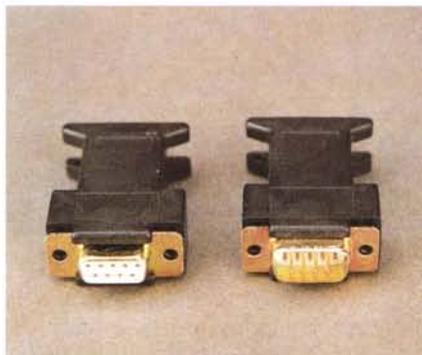
```
dir
vedremo comparire la lista di tutte le macchine collegate in rete come fossero tante directory dell'unità «net:». Analogamente possiamo listare una di queste directory nei due modi già indicati (posto che una macchina si chiami «platone»):
```

```
dir net:platone
oppure, procedendo con i cd:
cd platone
dir
```

in tutt'e due i casi vedremo la lista dei volumi disponibili sulla macchina platone, ancora come subdirectory. Possiamo continuare la nostra esplorazione



*Per collegare due sole macchine in rete attraverso l'interfaccia seriale è sufficiente un comunissimo cavo Null Modem tra le due macchine. Per collegare, di contro, più di due macchine è necessario dotare ogni nodo di uno «sdoppiatore passivo» di seriale: nella foto è mostrato il prototipo da noi utilizzato, mostrato frontalmente e posteriormente.*



*Sulla prima ed ultima macchina della rete è necessario chiudere l'anello con un apposito terminatore. In questo modo le due macchine «terminate» saranno una successiva all'altra.*

sulla rete, accedendo ad esempio all'hd della macchina platone per vedere cosa c'è dentro. Sempre con i due metodi daremo:

```
dir net:platone/dh0
oppure, continuando a spostare la nostra CurrentDir:
cd dh0 (attenzione: senza i «:» dal punto di vista nostro si tratta di una normale directory, non di un volume)
seguito da un «dir».
```

Così il path completo del file «pippo» contenuto nella directory «temp» dell'hd della macchina platone sarà:

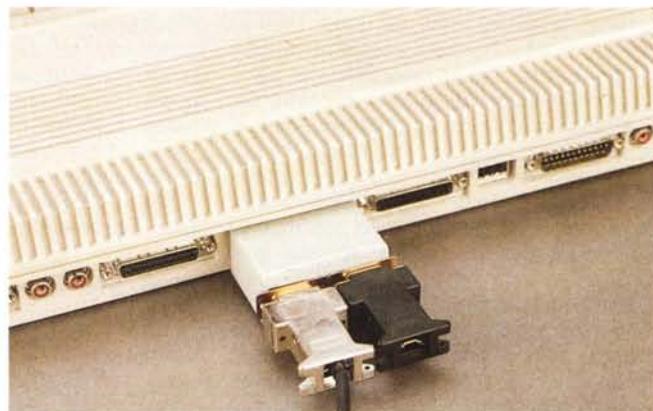
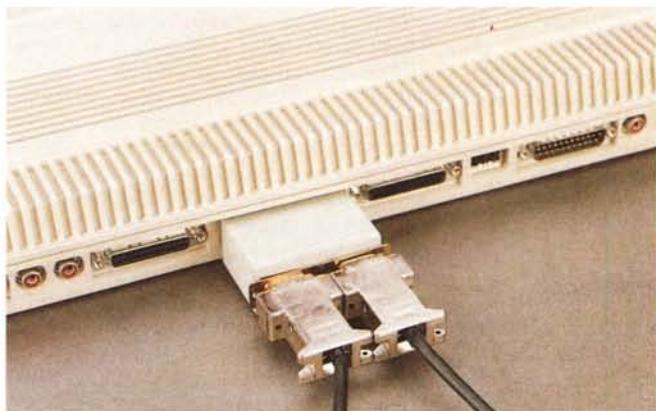
```
net:platone/dh0/temp/pippo
e per visualizzarlo sarà sufficiente scrivere
```

```
type net:platone/dh0/temp/pippo
o se avevamo proceduto a colpi di cd ed eravamo rimasti all'ultimo «cd dh0» basterà digitare
```

```
cd temp
type pippo
oppure:
type temp/pippo
insomma, come più ci aggrada.
```

Da ogni macchina possiamo addirittura aggiungere al nostro path di ricerca anche directory presenti su dischi remoti così come assegnare simboli «remoti». Facciamo un paio di esempi. Immaginiamo che nella macchina platone, o meglio nel suo hd, vi sia una nutrita serie di programmi che vogliamo «vedere» sempre anche dalla nostra postazione. Diciamo che questi programmi stanno nella directory «tools». Scrivendo sulla nostra:

```
path net:platone/dh0/tools add
ogni comando che digiteremo da quel momento in poi sarà cercato anche sulla macchina remota (nella directory tools) per essere eseguito. Analoga-
```



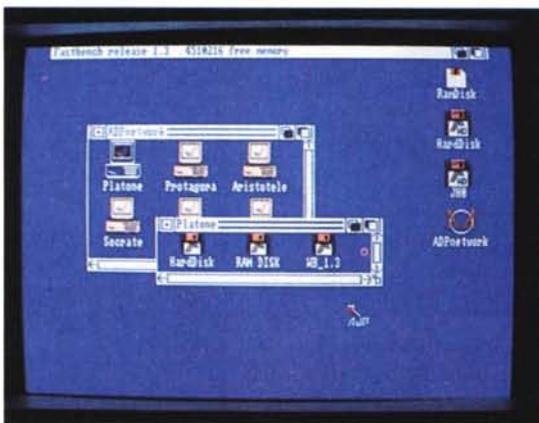
*Lo «sdoppiatore» passivo di seriale attaccato rispettivamente ad una macchina intermedia e ad una macchina estrema. Tale distinzione è però solo dispozionale in quanto a livello elettrico le macchine sono collegate tra loro attraverso una architettura circolare. Sul numero 90 di MC a pagina 219 abbiamo pubblicato lo schema elettrico degli sdoppiatori e terminatori collegati agli Amiga.*



Una volta installata la rete, appare sul WorkBench una nuova icona a forma di anello, come la struttura di ADPNetwork.



Clickando su quest'icona appaiono le macchine collegate, in quel momento, in rete.



Per conoscere quali dispositivi di memorizzazione sono disponibili sulle varie macchine è sufficiente clickare sulle relative icone. In questa foto vediamo i volumi disponibili sulla macchina platone.



Clickando su un'icona di un device remoto, accederemo a questo come fosse device della nostra macchina: a questo punto possiamo effettuare qualsiasi operazione ammessa sui file locali anche sui file remoti.

mente possiamo assegnare simboli:  
 assign c: net:platone/dh0/c  
 assegnerà la nostra directory dei comandi «c:» alla directory c dell'hd della macchina platone. Da quel momento in poi ogni comando eseguito da shell verrà prima caricato dalla macchina platone e poi eseguito sulla nostra macchina platone.

Così ogni programma, in grado di vedere device montati (la quasi totalità del sw esistente) sarà compatibile con la nostra rete: basta solo indicare, per i file remoti, path «di rete». Facciamo un esempio, chiamando a testimoniare il nostro fiore all'occhiello italiano «C1-Text». Montata la rete proviamo a leggere un file di testo dalla directory «s» della macchina pitagora col nostro C1-Text in esecuzione sulla macchina platone.

Caricato il programma, dal menu «Generale» scegliamo l'item «Aprire documento», come abbiamo sempre fatto. Prima novità: nel requester che appare, accanto alla scritta «unità di memoria», oltre ai normali device disponibili troviamo un nuovo button indicante «net»: è il device della rete. Col mouse clickiamo

su questo, e vedremo apparire la lista delle macchine in rete. Sempre col mouse clickiamo su pitagora: immediatamente appare la lista dei device di questa macchina. Clickiamo su dh0 e poi (non appena apparsa la directory di questo hd remoto) sulla directory «s». Siamo arrivati, scegliamo il nostro file (ad esempio la startup-sequence di quella macchina) e clickiamo sul gadget «Procedere». C1-Text non lo sa, ma sta caricando un file dalla rete. Con un procedimento assolutamente analogo possiamo salvare il file da qualsiasi altra parte, sull'intera rete.

È così con qualsiasi altro programma, di qualunque genere, riguardo ogni possibile tipo di file da leggere o salvare su rete. Semplice, no?

### Compatibilità e WorkBench

Tutto quello che abbiamo detto riguardo l'utilizzo della rete da shell o dall'interno di qualsiasi programma commerciale permane anche nell'utilizzo via WorkBench.

Infatti, non appena entrati in rete (operazione demandata alla startup-se-

quence del disco di boot o a seguito dell'attivazione di una apposita icona), anche il WB mostrerà il nuovo device «net» come uno dei dischi accessibili. Se proviamo a clickare sull'icona «net», si aprirà una finestra contenente tante icone a forma di computer quante sono le macchine collegate in rete in quel momento. Se in seguito nuovi nodi si aggiungeranno alla rete, automaticamente questi saranno visibili dalle restanti postazioni come nuove icone nella finestra del device «net».

Come visto prima per le operazioni di shell, clickando sull'icona di uno dei computer in rete vedremo tutti i device di quella macchina. A questo punto un accesso ad una qualsiasi di queste icone sarà un accesso al corrispondente device remoto. Così potremo copiare file da una macchina all'altra semplicemente spostando icone alla stessa maniera di come faremmo, su singola macchina, per spostare file da un device ad un altro. Analogamente clickando sull'icona di un qualsiasi programma presente su un device remoto avremo l'effetto di caricarlo e lanciarlo sulla nostra macchina. Addirittura possiamo cancellare,



Discorso del tutto analogo all'interno di applicazioni. In questo caso il WP italiano C1-Text mostra nel requester «Aprire Documento» un nuovo button «net» che ci permette di accedere alla rete. Clickando su questo appaiono le macchine in rete in quel momento.

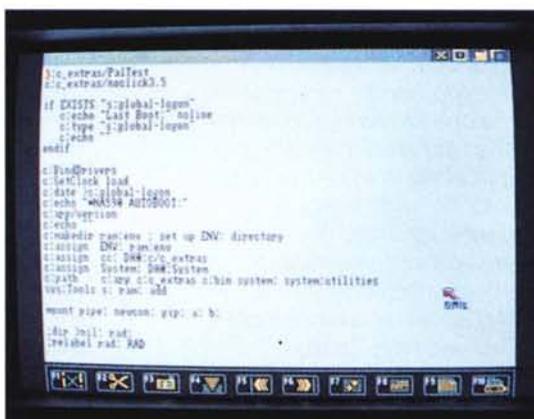


Accedendo alla macchina pitagora e poi all'HD di questo vedremo scorrere la lista dei file e directory presenti su quel disco remoto.



Scegliamo dalla directory s dell'HD remoto di pitagora il file «startup-sequence».

Pochi istanti dopo il file è arrivato sulla nostra macchina e trattato da C1-Text come un file qualsiasi. Da notare che il path di rete resta come parte integrante del nome del file, quindi salvando nuovamente il file senza modificare il nome (selezione «Memorizzare Stesso» del menu «Generale») otterremo di salvare nuovamente il file sul device remoto.



rinominare, duplicare e chiedere informazioni su file remoti.

L'unica cosa che non possiamo fare è formattare la rete...

### A tutti i programmatori: ADPnetwork e le applicazioni distribuite

ADPnetwork vera e propria è una collezione di processi che permette a qualsiasi programma in esecuzione su una macchina di dialogare con qualsiasi altro programma in esecuzione su un'altra macchina. In pratica i Net-Handler e i Net-Server di Ciuchini e Suatoni presenti su tutte le macchine collegate in rete utilizzano i processi di ADPnetwork per scambiarsi i Dos\_Packet di AmigaDOS più altri messaggi.

Analogamente è possibile scrivere qualsiasi altra applicazione distribuita utilizzando l'ultima versione dell'ADPmttb che è disponibile, su richiesta e gratuitamente, a chiunque volesse sviluppare applicazioni su rete.

Molto semplicemente, se un processo della macchina platone deve mandare un messaggio sulla porta «pippo» di

un processo in esecuzione sulla macchina pitagora sarà sufficiente utilizzare la funzione SendBlock con il suo nuovo «MODE\_NET» nel seguente modo: SendBlock(MODE\_NET, "pitagora/pippo", Msg, Len); dove «Msg» è il puntatore al messaggio da spedire e «Len» è la lunghezza dello stesso.

Complementarmente, il processo destinatario in esecuzione sulla macchina pitagora, oltre ad aver creato una porta MTTB di nome «pippo» attraverso la funzione «NewPort» effettuerà una semplice:

Len = ReceiveBlock(MODE\_WAIT, "pippo", Msg);

Da segnalare che per il processo destinatario non cambia nulla se il messaggio ricevuto arriva da un altro processo in esecuzione sulla medesima macchina, o da un processo remoto in esecuzione su un altro nodo: a «lui» è arrivato un messaggio e basta. Poi naturalmente, sarà cura del mittente indicare la vera provenienza all'interno della busta chiusa...

E allora perché non cominciare già a pensare ad animazioni in tempo reale

realizzate da più computer in rete, immagini ray tracing costruite parallelamente o, magari, a stupendi videogame multiutente con cui attrezzare vere e proprie sale gioco? Il limite, come sempre, la fantasia e l'inventiva di chi programma. Nulla di più.

### Concludendo

In questo viaggio all'interno di Amiga iniziato circa un anno fa ci siamo sicuramente divertiti un sacco. ADPnetwork voleva essere inizialmente solo un'esperienza didattica da rigirare a tutti i lettori «Amiga» di MC ma poi è diventata, forse casualmente, qualcosa di più. Ne abbiamo avuto riprova tanto durante lo SMAU dell'89 dove era mostrato un vero prototipo, quanto scambiando impressioni «tecniche» con gli altri sviluppatori Amiga durante l'ultima DevCon di Parigi. Non ci rimane allora che darvi appuntamento sul numero di settembre di MC per maggiori notizie sulla commercializzazione di ADPnetwork e al prossimo SMAU 90 per toccare finalmente con mano tutte le nostre parole di questi ultimi 10 mesi. Arrivederci...