

Questo mese il prode Mangrella ci propone una comoda utility per scambiare, con metodi assolutamente software, il drive interno e il drive esterno di Amiga. Ovviamente tale scambio resisterà anche ai boot successivi tant'è che sarà possibile perfino avviare la macchina da DF1. O, meglio, dal nostro nuovo drive esterno DF0. Provare per credere...

BootDF1

di Maurizio Mangrella - Eboli

Avete mai pensato che, da quando avete acquistato il più bel computer della vostra vita (sto parlando dell'Amiga, naturalmente), non avete fatto altro che effettuare il boot e caricare programmi in prevalenza dal drive interno? E avete mai pensato che, nonostante la proverbiale affidabilità dei drive dell'Amiga (beh, in verità non tanto proverbiale: ma il mio resiste da oltre due anni e non ha mai «sgarrato» di un bit), un giorno o l'altro il vostro drive interno potrebbe abbandonarvi? (Per tutti, prima o poi, giunge il momento di tirare le cuoia... anche per i computer!).

Per chi ha il drive esterno il problema esiste solo in parte: l'uso prolungato, continuativo e notturno di questo (o di questi, per quei fortunati che ne hanno due o tre) può allungare sensibilmente la durata del drive interno, ormai adoperato solo per il boot e per le copie.

Tuttavia sarebbe assai più «papabile» una soluzione (software! Potete anche non crederci, ma è così!) che consenta di far assurgere il DF1: al ruolo di «drive interno» e trasformi, invece, la meccani-

ca built-in in DF1:. Se non avete capito il gioco di parole, spiego in termini più pratici: dopo aver lanciato il mio BootDF1 e aver resettato l'Amiga il drive esterno (dovete averlo!) diventa DF0: e quello interno DF1:, per cui sarà il drive esterno a clickare e ad effettuare il boot. Contenti?

Non ci credo!

Il principio sul quale si basa BootDF1 è, al solito, molto semplice: al momento dell'esecuzione BootDF1, se non specificato altrimenti sulla linea di comando (v. dopo) si installa in memoria a partire dall'indirizzo \$7EC00 (vedremo dopo il perché di questa scelta) e si insedia tra i moduli Resident del sistema operativo. Dopo il reset, l'«appendice» residente di BootDF1 modifica la routine OpenDevice() della exec.library in modo da intercettare tutte le chiamate alla trackdisk.device (che è, come noto, il device utilizzato dall'S.O. per la gestione dei floppy). La nuova routine si limita a controllare il nome del device richiesto e l'unità alla quale si fa riferimento: se il nome è 'trackdisk.device' e l'unità è minore di 2 — ossia è 0 (drive interno) o 1 (drive esterno) — allora la routine trasferisce il controllo alla vec-

```
* BootDF1.asm          by Maurizio Mangrella 1989
*
* Questo programma e' una soluzione software al problema dell'usura
* del drive interno dell'Amiga, e permette di effettuare il boot dal
* DF1: (il primo drive esterno della catena). Dispone di un help e
* di una possibilita' di richiesta di informazioni.
* Una volta installato, i due drives (quello interno e il primo
* degli esterni) sono scambiati (per cui DF0: sara' il drive esterno
* e DF1: quello interno).
*
* Qualche definizione utile
NT_UNKNOWN      EQU    0
RTW_COLDSTART   EQU    1
MEMF_PUBLIC     EQU    1
_SysBase        EQU    4
*
* Riferimenti esterni
XREF             _LV0OpenLibrary
XREF             _LV0Output
XREF             _LV0Write
XREF             _LV0SumKickData
XREF             _LV0CloseLibrary
XREF             _LV0AllocMem
XREF             _LV0OpenDevice
```

(continua a pag. 226)

chia OpenDevice() scambiando i numeri delle unità; in altre parole, una OpenDevice come la seguente:

```
OpenDevice("trackdisk.device",1L,&myReq,0L);
```

che fa riferimento al drive esterno, sarà tradotta in

```
OpenDevice("trackdisk.device",0L,&myReq,0L);
```

facente invece riferimento al drive interno. A patto che il programma chiamante non effettui un controllo sulla struttura PublicUnit — ma in fin dei conti questa eventualità è assai rara — non si accorgerà mai dell'imbroglio. Il DOS, come avrete capito, è il primo che viene imbrogliato.

Se, invece, si fa riferimento ad altre unità (DF2: o DF3:) o ad un altro device, la routine trasferisce il controllo alla vecchia OpenDevice() senza ulteriori modifiche.

Uso del programma

BootDF1 presume di essere SEM-PRE lanciato da CLI; nel caso venga lanciato da WorkBench, i risultati sono imprevedibili.

BootDF1 presenta le seguenti sintassi:

```
BootDF1 ?      Mostra un piccolo help e le
                sintassi da CLI
BootDF1       Mostra il drive attivo per il
DRIVE:D:d     boot
BootDF1       Installa o disinstalla il pro-
                gramma.
```

BootDF1, lanciato così com'è, provvede la prima volta a installarsi e la seconda a disinstallarsi: ogni volta, insomma, fa il contrario della volta precedente. Per valutare quale sia effettivamente il drive attivo per il boot, BootDF1 controlla la coerenza di alcuni dati che lascia in memoria (v. dopo).

Dal punto di vista della versatilità, ricordo che esistono fondamentalmente due modi per accedere a un drive: il DMA hardware e i device (in questa categoria dobbiamo contemplare anche i vari DF0:, DF1:, etc., del DOS). Contro il DMA hardware (utilizzato in molti caricatori veloci di giochi commerciali) il programma, come avrete intuito, non può assolutamente nulla; nel caso dei device, invece, è assolutamente efficace.

Ai «pirati» ricordo che molti giochi sprotetti fanno uso della trackdisk.device per caricarsi: quindi con BootDF1 non ci sono problemi.

Il programma

Per motivi di versatilità e efficienza nella gestione dei moduli Resident, utili-

(segue da pag. 225)

```
* Qui comincia il programma...

Main      MOVE.L  A0,A5                ;salva l'ind. degli argomenti
          MOVE.L  _SysBase,A6        ;apre la dos.library
          LEA    DOSName(PC),A1      ;forma l'indirizzo del nome
          CLR.L  D0                    ;qualunque versione va bene
          JSR   _LVDDOpenLibrary(A6) ;apre la libreria
          MOVE.L D0,DOSBase         ;e ne conserva il puntatore

          MOVE.B (A5),D0            ;preleva il primo carattere
          ;della linea di comando
          CMPI.B #'?',D0             ;e' un '?' ?
          BEQ   HelpOnLine          ;si': mostra l'help
          ORI.B  #$20,D0             ;setta il bit 5 (e trasforma
          ;le maiuscole in minuscole)
          CMPI.B #'d',D0            ;e' una 'd' ?
          BEQ   ShowDrive           ;si': mostra il drive attivo
          CMPI.L #'BOOT',#7EC22     ;il BootDF1 e' installato?
          BNE.S Install            ;no: lo installa
          ;(il programma avverte di
          ;essere stato installato se
          ;trova la parola chiave 'BOOT'
          ;all'indirizzo #7EC22)

DeInstall  CLR.L  #226(A6)          ;cancella la lista dei moduli
          ;Resident aggiuntivi
          JSR   _LVDSumKickData(A6) ;ricalcola il checksum
          MOVE.L D0,#22A(A6)        ;e lo pone in KickChecksum
          CLR.L  #7EC22              ;cancella la keyword 'BOOT'
          MOVE.L DOSBase(PC),A6     ;preleva DOSBase

          JSR   _LVDDOutput(A6)      ;determina l'Output File Handle
          MOVE.L D0,D1
          LEA   DIMessage(PC),A0     ;scrive il messaggio di dein-
          ;stallazione
          MOVE.L A0,D2
          MOVEQ #44,D3
          JSR   _LVDDWrite(A6)       ;scrive il messaggio
          BRA   Ending              ;e termina

Install    MOVE.L  #0007EC00,A1      ;forma l'indirizzo della Resident
          LEA   Resident(PC),A0     ;prepara i puntatori per la copia
          MOVEQ #16,D0               ;17 words (34 bytes) da copiare
          MOVE.W (A0)+,(A1)+        ;copia, copia...
          DBRA D0,Loop1

          MOVE.L #0007EC26,A1        ;a #7EC26 comincia il programma
          LEA   Routine(PC),A0      ;prepara i puntatori per copiare
          MOVEQ #RoutineSize,D0     ;la routine di installazione
          MOVE.W (A0)+,(A1)+        ;a' ricopia, ricopia...
          DBRA D0,Loop2

          MOVE.L #0007EC00,#226(A6) ;pone l'indirizzo della lista
          ;dei moduli Resident in
          ;KickTagPtr
          JSR   _LVDSumKickData(A6) ;ricalcola il checksum
          MOVE.L D0,#22A(A6)
          MOVE.L #'BOOT',#7EC22     ;scrive la keyword 'BOOT'

          MOVE.L DOSBase(PC),A6     ;scrive il messaggio di instal-
          JSR   _LVDDOutput(A6)      ;lazione sull'Output Stream
          MOVE.L D0,D1
          LEA   INMessage(PC),A0    ;termina ordinatamente
          MOVE.L A0,D2
          MOVEQ #41,D3
          JSR   _LVDDWrite(A6)
          BRA   Ending

ShowDrive  LEA   DRMessage(PC),A5   ;forma l'indirizzo del messaggio
          CMPI.L #'BOOT',#7EC22     ;controlla l'avvenuta instal-
          BNE.S Skip1               ;lazione: se si' aggiusta il DF0:
          ADDQ.B #1,28(A5)          ;in DF1: (incrementando il codice
          ;ASCII)
          MOVE.L DOSBase(PC),A6     ;stampa il messaggio
          JSR   _LVDDOutput(A6)
          MOVE.L D0,D1
          MOVE.L A5,D2
          MOVEQ #31,D3
          JSR   _LVDDWrite(A6)
          BRA   Ending              ;ed esce

HelpOnLine MOVE.L  DOSBase(PC),A6   ;l'help: vengono mostrate tutte
          JSR   _LVDDOutput(A6)     ;le funzionalita' del programma
          MOVE.L D0,D1
          LEA   HLMessage(PC),A0
          MOVE.L A0,D2
          MOVE.L #19B,D3
          JSR   _LVDDWrite(A6)
```

```

Ending      MOVE.L  A6,A1          ;preleva DOSBase in A1
            MOVE.L  _SysBase,A6      ;una routine di Exec
            JSR    _LVDCloseLibrary(A6) ;chiude la dos.library
            CLR.L  D0              ;ritorna 0 (RETURN_OK)
            RTS                    ;esce

Routine     MOVEM.L  D0-D1/A0-A1/A4-A6,-(SP) ;salva i registri
            MOVE.L  _SysBase,A6      ;preleva _SysBase
            LEA    _LV00OpenDevice+2(A6),A5 ;forma l'indirizzo del vettore
            LEA    OldOpDev+2(PC),A4  ;della funzione OpenDevice()
            LEA    OldOpDev+2(PC),A4  ;preleva l'indirizzo della
            MOVE.L  (A5),(A4)         ;longword da correggere
            MOVEQ  #NewOpDevSize,D0  ;salva il vecchio valore
            MOVEQ  #MEMF_PUBLIC,D1   ;grandezza della NewOpDev
            JSR    _LV0AllocMem(A6)  ;memoria accessibile a tutti
            MOVE.L  D0,A1            ;alloca un po' di memoria
            LEA    NewOpDev(PC),A0    ;prepara per la copia
            MOVEQ  #NewOpDevSize,D1  ;copia NewOpDevSize bytes
            MOVE.L  (A0)+,(A1)+      ;a' stracopia, stracopia...
Loop4       DBRA    D1,Loop4
            MOVE.L  D0,(A5)         ;salva il nuovo indirizzo nel
            MOVEM.L (SP)+,D0-D1/A0-A1/A4-A6 ;vettore della OpenDevice()
            RTS                    ;recupera i registri ed esce

NewOpDev    MOVEM.L  D0-D1/A0-A1,-(SP) ;salva i registri
            LEA    TDName(PC),A1     ;forma l'indirizzo del nome
            Loop5   TST.B  (A1)       ;della trackdisk.device
            BEQ.S  Found1           ;il byte e' '\0'?
            CMPM.B (A0)+,(A1)+      ;si': e' la trackdisk.device
            BEQ.S  Loop5            ;no: confronta i bytes corrisp.
            MOVEM.L (SP)+,D0-D1/A0-A1 ;se sono uguali continua
            JMP    #00111111        ;niente da fare: recupera i
Found1      CMPI.W  #2,D0           ;registri ed esce
            BGE.S  NotFound         ;se il numero del drive richie-
Found2      MOVEM.L (SP)+,D0-D1/A0-A1 ;sto e' maggiore di 1 esce
            EDRI.W  #1,D0           ;riprende i registri...
            BRA.S  OldOpDev         ;scambia i numeri dei drives
            ;ed esce

TDName      DC.B    'trackdisk.device',0 ;nome della trackdisk.device
            CNOP   0,2

EndNewOpDev EQU    EndNewOpDev-NewOpDev ;grandezza della NewOpDev
NewOpDevSize

EndRoutine  EQU    EndRoutine-Routine  ;grandezza della Routine
RoutineSize

* Area dati

DOSName     DC.B    'dos.library',0
            CNOP   0,2
DOSBase     DC.L    0

Resident    DC.L    #0007EC08
            DC.L    0
            DC.W    #4AFC
            DC.L    #0007EC08
            DC.L    #0007EC22
            DC.B    RTW_COLDSTART
            DC.B    33
            DC.B    NT_UNKNOWN
            DC.B    -5

            DC.L    #0007EC1A
            DC.L    0
            DC.L    #0007EC26

DIMessage   DC.B    'Resettare l''Amiga per disinstallare BootDF1',10,0
INMessage   DC.B    'Resettare l''Amiga per installare BootDF1',10,0
DRMessage   DC.B    'Drive attivo per il boot: DF0:',10,0
HLMessage   DC.B    27,'[ImBootDF1',27,'[0m by Maurizio Mangrella'
            DC.B    ' 1989',10,10
            DC.B    'USD: BootDF1 <DRIVE> Mostra il drive attivo',10
            DC.B    '      BootDF1 <?> Mostra questo help',10
            DC.B    '      BootDF1 Installa o disinstalla il'
            DC.B    ' programma',10,0

            END                    ;vi e' piaciuto?

```

Sorgente commentato di BootDF1.

ty come queste vengono sempre scritte in Assembler; BootDF1 (che spero di aver commentato in maniera sufficientemente abbondante) è comunque comprensibile anche a chi è completamente digiuno di L.M. (garantito!).

A seconda del primo carattere della linea di comando passata dal CLI al programma (l'indirizzo della command line è in A0 e la lunghezza, «newline» finale compreso, è in D0) il programma salta a HelpOnLine (per la visualizzazione dell'help), a ShowDrive (per il drive correntemente attivo), a Deinstall (per de-installarsi dalla memoria) e a Install (per installarsi, invece).

Il programma, come già detto più volte, si installa a partire dall'indirizzo \$7EC00: piuttosto basso, rispetto al top della memoria chip (che è \$7FFFF), ma sicuro contro eventuali invasioni. La più pericolosa di queste ultime è il modulo residente del device RAD:, che si insedia intorno a \$7FF30 e che distrugge la maggior parte dei moduli residenti di altri programmi.

Chi si intende di Assembler e di moduli residenti avrà notato che la Routine (che, al di là del poco fantasioso nome, si incarica di rimanere residente e di installare la nuova OpenDevice()) alloca un po' di memoria per mantenervi la NewOpDev (il sostituto della vecchia OpenDevice()): ciò sarebbe inutile, dal momento che il modulo Routine è per definizione residente (dunque non ha bisogno di lasciare copie di sue parti sparse per la memoria), ma è più sicuro nel caso di un programma «invadente» che inopinatamente cancelli la parte alta della chip RAM. Se non si provvedesse ai ripari in tal senso, una Guru Meditation sarebbe in agguato entro una decina di minuti.

In caso di cancellazione del programma dalla memoria, il S.O. provvede automaticamente a eliminare il modulo Resident di BootDF1 dalla lista Kick-TagPtr: in tal caso sarà sufficiente ripetere l'installazione.

Per controllare se il programma si è installato o meno, BootDF1 controlla se all'indirizzo \$7EC22 è presente la parola chiave 'BOOT': se, anche dopo una installazione, BootDF1 si ostina a dire che il drive attivo per il boot è DF0:, è consigliabile ripetere l'installazione.

Tutti gli output del programma, infine, sono resi con le routine del DOS Output() e Write(): più semplice di quanto possa sembrare.

Vi saluto augurandomi che il programma vi torni utile.