

I «device driver»

quarta parte

Proseguiamo in questa puntata l'analisi delle funzioni di un «device driver» e cioè di quelle funzioni che un driver che si rispetti dovrebbe implementare.

Parlando di driver ci riferiremo sia ad un ipotetico driver già esistente sia ad un driver che noi dobbiamo implementare: nel primo caso è come se si analizzasse se il driver in esame possiede tutti i requisiti, mentre nel secondo caso è come se ci imponessimo via via delle regole che il driver in costruzione dovrà seguire.

La scorsa puntata abbiamo iniziato conoscendo la funzione INIT, di fondamentale importanza in quanto è sempre la prima routine del driver che viene attivata, ed analizzando la struttura detta BPB («BIOS Parameter Block»). Proseguiamo dunque con le successive funzioni, facendo riferimento alla figura 1, che continueremo a riproporre in ogni puntata in quanto veramente utile

La funzione «MEDIA CHECK»

Questa funzione serve al driver per riconoscere il tipo di dispositivo con cui ha a che fare, nel caso in cui il dispositivo in esame gestisca quantità di tipo «block»: in particolare si fa riferimento ad un byte, fornito dal DOS, detto «media descriptor byte» (vedasi la figura 3), che specifica le caratteristiche fondamentali del dispositivo.

Dalla figura 3 vediamo che gli unici tre bit che possono essere diversi da 1 sono i tre meno significativi ed hanno il seguente significato:

- "a" = 0 se il dispositivo è di tipo rimovibile (ad es. un floppy).
- "a" = 1 se viceversa si tratta di un dispositivo fisso (un hard disk)
- "b" = 0 se il dispositivo è dotato di un numero di settori differente da 8, mentre
- "b" = 1 se i settori sono 8
- "c" = 0 se il dispositivo non è «a 2 facce» (un hard disk non ha 2 facce, ma in genere ne ha di più, a seconda del numero di testine) mentre
- "c" = 1 se il dispositivo ha «2 facce».

Per le versioni di DOS che vanno fino alla 3.1 (delle successive versioni e degli eventuali cambiamenti ci occuperemo nel seguito) abbiamo i seguenti valori del «descriptor byte», per le unità più diffuse:

valore	dispositivo
F8H	hard disk
F9H	3".5 da 720k (2 lati, 9 settori, 80 tracce)
F9H	5".25 da 1.2M (2 lati, 15 settori, 80 tracce)
FDH	5".25 da 360k (2 lati, 9 settori, 40 tracce)

Tutto sommato balza subito agli occhi un'incongruenza data dal fatto che i due dispositivi aventi descriptor pari a F9H sembrerebbero trattarsi di dispositivi non rimovibili, dal momento che il bit «a» è nullo: mistero!

Tra l'altro sono previsti pure i valori FDH ed FEH relativi alle «mitiche» unità a dischi da 8", che francamente non abbiamo mai visto connesse ad un PC sotto MS-DOS: quanto sono più maneggevoli e sicuri i microfloppy da 3", a confronto dei flessibilissimi 8" (da cui deriva il nome di «floppy»...), senza contare che i primi si possono portare agevolmente nel taschino della camicia, mentre gli altri richiedono mastodontiche scatole...

Tornando dunque alla funzione «MEDIA CHECK», abbiamo detto che il DOS fornisce nel Request Header (riportato in figura 2) il descriptor del dispositivo di cui si vuole testare lo stato: in particolare il driver dovrà rispondere alla chiamata della funzione fornendo due valori in altrettanti campi dell'header. Dovrà fornire un «Return Byte», i cui unici valori e relativi significati sono i seguenti:

- un valore pari ad FFH (pari a -1) indicherà che il «media», il supporto, è stato cambiato (ad esempio noi abbiamo estratto un floppy per sostituirlo con un altro);
- un valore nullo indica che il driver non è in grado di sapere se il supporto è stato cambiato (ben strano...);
- un valore, infine, pari a 01H indica

codice	funzione	
	block device	character device
00	INIT	INIT
01	MEDIA CHECK	NOP
02	BUILD BPB	NOP
03	TOCTH input	TOCTH input
04	INPUT	INPUT
05	NOP	NONDESTRUCTIVE INPUT
06	NOP	INPUT STATUS
07	NOP	INPUT FLUSH
08	OUTPUT	OUTPUT
09	OUTPUT with VERIFY	OUTPUT with VERIFY
0A	NOP	OUTPUT STATUS
0B	NOP	OUTPUT FLUSH
0C	TOCTH output	TOCTH output
0D	DEVICE OPEN	DEVICE OPEN
0E	DEVICE CLOSE	DEVICE CLOSE
0F	REMOVABLE MEDIA	BPB

Figura 1 - Come di consuetudine accompagnamo ogni puntata con la tabella, che suddivide tutte le funzioni di un driver in funzioni di tipo «block» o di tipo «character», cioè a seconda dei dati sui quali operano.

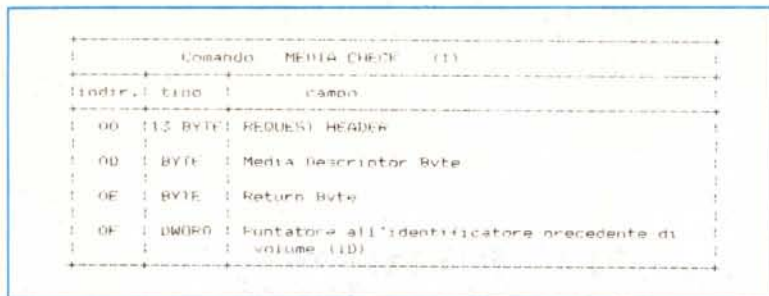


Figura 2 - Struttura del REQUEST HEADER nel caso di chiamata alla routine MEDIA CHECK, che serve ad identificare quale tipo di dispositivo («media» appunto) viene gestito dal driver stesso. Sul significato dei vari campi rimandiamo al testo, mentre in particolare per il campo «media descriptor» rimandiamo alla figura 3.

viceversa con certezza che il supporto non è stato cambiato.

Per quanto riguarda il secondo valore che il driver deve fornire, si tratta in realtà di un'informazione da gestire solo nel caso di versioni di DOS dalla 3.0 in su, come dire che un driver che gira solo sotto 2.0 non dovrà preoccuparsi di questa ulteriore informazione, non gestita appunto dalle versioni «più basse» e più vecchie: c'è da dire che ormai conviene lavorare con versioni di DOS a partire dalla 3.1, se non altro per le novità introdotte con quella versione, per poi passare obbligatoriamente alla 3.3 laddove il sistema preveda dei microfloppy.

Dicevamo dunque che la seconda informazione da fornire a cura del driver è il puntatore al campo di «identificazione del volume» relativo al dispositivo precedentemente in uso, in modo che il DOS possa verificare se il dischetto attualmente in uso abbia lo stesso identificatore del precedente potendo così arguire, in caso di differenza, che il dischetto è stato cambiato, indipendentemente dal valore fornito dal driver come «Return Byte».

Nel caso in cui il driver gestisse un'unità logica non dotata di identificatore di volume, ecco che allora sarà sua cura fornire al DOS il puntatore ad una stringa pari a «NO NAME» e terminante con un byte nullo.

La funzione BUILD BPB

Si tratta di un'altra funzione prettamente dedicata ai dispositivi di tipo «block» e serve ad aggiornare il BPB al quale il DOS farà successivo riferimento.

Prima di procedere facciamo un passo indietro: il BPB (che ricordiamo sta per «BIOS Parameter Block») non è altro che una tabella, di cui abbiamo parlato la scorsa puntata, che contiene una serie di informazioni utilissime al DOS per la corretta gestione del dispositivo.

Ricordiamo in breve che nel BPB sono presenti le seguenti informazioni:

- il numero di byte per settore;
- il numero di settori che compongono un'unità di allocazione («allocation block»);
- il numero di settori riservati;
- il numero di FAT presenti nell'unità;
- il numero massimo di voci consentite nella directory;
- il numero totale di settori;
- il «media descriptor byte» che ora conosciamo bene; ed infine
- il numero di settori occupati da una singola replica della FAT.

Ecco perciò che il BPB è in pratica la «carta d'identità» del dispositivo in uso e conseguentemente deve essere modificato allorché il driver, alla chiamata della funzione MEDIA CHECK, che il «media» è stato cambiato oppure che «non è sicuro».

Mentre nel caso di un personal dotato di due floppy da 5" tutto sommato tale funzione sembrerebbe ridondante, viceversa tale chiamata è di fondamentale importanza nel caso di un AT (vero o compatibile...) che come noto è dotato di un'unità a floppy da 5" a quadrupla densità, che può leggere indifferente dischetti da 360k oppure da 1.2M, in maniera trasparente, cioè senza che l'operatore debba comunicare al sistema operativo alcunché. Tra l'altro non è detto che l'operatore sappia se due dischetti hanno o meno la stessa capacità...

Ecco che così «per magia» il DOS sa in ogni caso con quale unità ha a che fare: ma a parte magheggi, come fa?

È presto detto: il «boot sector» di ogni disco deve contenere una serie di informazioni, tra le quali proprio il BPB.

Tra parentesi diciamo che il «boot sector» non è altro che il primissimo settore della prima traccia del disco e contiene, oltre alle informazioni salienti che tra breve analizzeremo, il cosiddetto «loader» e cioè la routine che caricherà da disco il sistema operativo: è infatti il primo settore che viene letto a seguito di un «reset software», ottenuto con la contemporanea pressione dei tasti «Ctrl», «Alt» e «Del» (il quale ultimo è diventato un orripilante «Canc»

nelle tastiere italiote: in genere è d'uso nominare tale tasto con la seconda «c» dura...). Tornando dunque al boot sector abbiamo che esso DEVE contenere le seguenti informazioni:

- un salto alla routine «loader», in un campo formato da tre byte: il primo è il codice EBH dell'istruzione «JMP», il secondo è il «displacement», utile all'Assembler per determinare l'indirizzo iniziale delle routine di loader ed il terzo byte deve essere un 90H, corrispondente all'istruzione NOP;
- il nome e la versione eventualmente customizzata del DOS, il tutto in 8 byte al massimo;
- il BPB (eccolo qui dunque...);
- una word rappresentante il numero di settori per traccia;
- una word che dice quante testine



Figura 3 - Ecco il cosiddetto «media descriptor byte» i cui tre bit che possono variare (almeno fino alla versione 3.1 dell'MS-DOS) sono stati indicati con «a», «b» e «c»: rimandiamo al testo per le varie combinazioni possibili.

sono presenti per la particolare unità (nel caso degli hard-disk);

- un'ultima word che indica il numero di settori nascosti «hidden» presenti nell'unità.

Tornando dunque alla funzione in esame, per la quale facciamo riferimento alla figura 4, abbiamo che il DOS fornisce al driver tanto il valore del «media descriptor byte» quanto quello del «Transfer Address»: il primo lo conosciamo già, perciò non aggiungiamo nulla.

Il secondo è invece l'indirizzo completo (sotto forma di «segment:offset») di un'area di memoria utilizzata come buffer per la lettura/scrittura di dati da e verso il dispositivo gestito dal driver stesso: non diciamo altro a riguardo di questo buffer di memoria, in quanto viceversa dovremmo introdurre alcuni nuovi concetti, che viceversa appesantirebbero notevolmente l'argomento.

Le funzioni INPUT ed OUTPUT

Abbiamo accomunato in un unico paragrafo due routine evidentemente antitetiche, in quanto hanno fondamentalmente la stessa struttura logica già a partire dal REQUEST HEADER e solo nella loro esecuzione si differenzieranno

nel primo caso ricevendo caratteri o settori (rispettivamente a seconda se il dispositivo è di tipo «character» oppure di tipo «block») oppure, ovviamente, inviando caratteri o settori, a seconda del tipo di dispositivo che gestiscono.

Per questo motivo d'ora in poi parleremo genericamente di byte o di settori da «gestire» sottointendendo che saranno da «inviare» o da «ricevere» a seconda se si tratterà di routine di OUTPUT oppure di INPUT.

Inoltre in un sol colpo analizzeremo

Comando			INDIR. BPB	(?)
indir.	tipo	campo		
00	12 BYTE	REQUEST HEADER		
0D	BYTE	Media Descriptor Byte		
0E	DWORD	Transfer Address		
12	WORD	Puntatore di byte (character device) o settori (block device)		
14	WORD	Start sector (solo per block device)		
16	DWORD	Puntatore al "volume identifier"		

Figura 5 - Struttura del REQUEST HEADER nel caso di chiamate a varie routine di INPUT e di OUTPUT: ferma restando la struttura della tabella, ovviamente il driver nelle varie situazioni si comporterà in maniera differente. Per quanto riguarda i «vari valori», diciamo che tale tabella è valida per cinque funzioni, due di INPUT e tre di OUTPUT.

Figura 4 - Struttura del REQUEST HEADER nel caso di chiamata alla routine BUILD BPB, che serve a ricostruire la tabella BPB in determinate situazioni, che vengono analizzate nel testo.

Ad esempio se il driver gestisce un circuito integrato chiamato USART («Universal Synchronous/Asynchronous Receiver Transmitter») per la rice/trasmmissione di caratteri attraverso una linea seriale, ecco che una stringa di IOCTL potrebbe essere quella che serve a settare la «baud rate» ed in generale le caratteristiche della comunicazione seriale: si badi bene dunque che questi caratteri non devono assolutamente essere inviati, ma viceversa sono necessari laddove ad esempio da programma si vogliono alterare le caratteristiche della comunicazione, passando da una baud rate ad un'altra.

Per quanto riguarda l'INPUT con IOCTL, (a proposito, ricordiamo che «IOCTL string» sta per «I/O ConTrol string») e cioè «stringa di controllo dell'I/O») è ovvio che si tratterà di una stringa che indicherà in quale stato si trova il dispositivo e, proseguendo nel nostro esempio, fornirà l'indicazione della baud rate attuale.

Infine per quanto riguarda la funzione 9, relativa all'OUTPUT con verifica, c'è semplicemente da aggiungere che sarà il driver a gestire dapprima un'operazione di OUTPUT ed una successiva operazione di INPUT per verificare che il tutto sia stato effettuato correttamente.

Come si vede non si tratta di una funzione particolarmente complessa, ma è ovviamente da prevedere laddove ci sia una sospetta od effettiva criticità nell'OUTPUT di informazioni verso un dispositivo: in questo ambito riteniamo di poter annoverare la «scrittura su disco», dal momento che può comportare parecchie volte dei problemi all'operatore.

Con questo terminiamo la puntata e rimandiamo alla prossima, dove termineremo di trattare l'analisi delle funzioni di un «device driver» e già fin d'ora promettiamo di analizzare nel dettaglio un driver già esistente e per certi versi ben noto, tanto per vedere applicate nella pratica le varie nozioni fin qui esaminate.

ben 5 routine, due per l'INPUT (le funzioni 3 e 4) e tre per l'output (le funzioni 8, 9 e OCH): secondo quanto detto prima accomuneremo le funzioni 4 ed 8 (semplici INPUT ed OUTPUT), poi le funzioni 3 e OCH (INPUT ed OUTPUT di stringhe IOCTL) per lasciare da sola la funzione 9 (OUTPUT con verifica).

Vediamo perciò le due funzioni «generiche» di INPUT e di OUTPUT:

in particolare si riferiscono alla gestione tanto di dispositivi di tipo «character» che di tipo «block».

Facendo riferimento alla figura 5, troviamo ancora il «media descriptor byte», che ovviamente serve come il pane nel caso di dispositivi di tipo «block» mentre nell'altro caso è perfettamente inutile; comune ad entrambe le categorie di «device» è viceversa il campo successivo, il «Transfer Address», il quale altro non è che il puntatore (del tipo «offset:segment») al buffer di memoria contenente le informazioni da gestire, sia che si tratti di caratteri (ed allora il driver gestirà un suo puntatore «all'interno» del buffer stesso) sia che si tratti di settori (nel qual caso può essere che nel buffer sia presente più di un settore).

Il campo successivo, «byte count» per i caratteri e «sector count» per i «block device», è di fondamentale importanza in quanto servirà al driver a segnalare al DOS quanti caratteri o settori ha gestito effettivamente: per quanto riguarda i settori ecco pronto il cam-

po successivo, che indicherà quale è il numero del primo settore da gestire.

Quest'ultimo campo ovviamente non ha significato per la gestione di caratteri, anche se forse si sarebbe potuto sfruttare come puntatore al buffer di caratteri da inviare uno alla volta, invece di far gestire al driver il puntatore «interno» di cui abbiamo parlato prima.

Infine l'ultimo campo (per le versioni di DOS dalla 3.0 in poi) serve al driver per comunicare al DOS il puntatore ad una stringa (terminante con un byte nullo) che rappresenta l'identificatore di volume del dispositivo da gestire, nel caso che quest'ultimo sia stato trovato cambiato rispetto all'ultima operazione (ad esempio se l'utente ha cambiato il dischetto su cui si stava lavorando): in tal caso il driver deve comunicare al DOS, oltre a tale puntatore, anche un codice d'errore pari a OFH, che ricordiamo rappresentare un «invalid disk change».

In tal modo il DOS provverà a emettere un messaggio che invita l'utente a reinserire il disco corretto.

Per quanto riguarda le funzioni che gestiscono le cosiddette «IOCTL string», c'è innanzitutto da ricordare che si tratta di particolari sequenze di caratteri che non sono da gestire come caratteri «normali», ma che viceversa servono, in OUTPUT, a settare il dispositivo in uno stato predeterminato ed in INPUT a conoscere lo stato in cui si trova il dispositivo.



ANKO MOUSE

In omaggio PAD MOUSE

L. 59.000 + IVA

ANKO MOUSE E PAD MOUSE IN OMAGGIO SONO DISPONIBILI PRESSO:

ALL COMPUTERS SRL, Via Cacciatori delle Alpi 25/A, SPOLETO • ANLISYS SRL, Via Cintia P.co S. Paolo, NAPOLI • ARIS SAS di Giampiero Zonni, Via Toselli 13, BRESCO (MI) • ARMONIA di Mariotto e Vanzella SNC, V.le Carducci 5, CONEGLIANO (TV) • ARTWORK ENGINEERING SAS, Vico A. Magnasco 5, GENOVA • BUCCI FRANCO, Corso Tacito 76, TERNI • BUSTO BIT di Vittorio Ornago SRL, Via Goviniana 17, BUSTO A. (VA) • CAD GRAPHICS SRL, Via Giusti 3, PADOVA • CDK SYSTEM, V.le Lincoln trav. 32-34, CASERTA • CIMA ELETTRONICA SAS, Via L. da Vinci 7, LECCO (CO) • COMMODORE CLUB CAMPANIA, Via Port'Alba 17/A, NAPOLI • COMMODORE CLUB CAMPANIA, Via Costantinopoli 19, NAPOLI • COMPUTER SOFT CENTER, Via Simeone 15, SIRACUSA • COOPERATIVA UNIVERSITARIA STUDIO E LAVORO, Piazza Lenardo da Vinci 32, MILANO • COPYSTAR, Via Fontana 34, SOMMA LOMBARDO (VA) • DATA PROGRESS SRL, Via Fiume 13, VIMERCATE (MI) • E.C. COMPUTERS SNC, Via S.ta Maria delle Grazie 4, PERTOSA (SA) • ECO INFORMATICA, Via Picicelli 21/25, BENEVENTO • FLOPPERIA SRL, V.le Montenero 31, MILANO • FOTONOVA di Bianchi Rita & C. SNC, Via Valeriano 1, SAN PIETRO BERBENNO (SO) • GE.COM. srl, Via ASiogo 17, TAVERNOLA (CO) • GENERAL COMPUTERS SAS, Via Bernini 101, NAPOLI • GREEN DATA SNC, Via S. Cristoforo 34, PORTICI (NA) • GUERRA EGIDIO & C. SAS, Via Bisuola 20/A, MESTRE • INFOBITSHOP SRL, Via S. Leonardo 120, SALERNO • INFOWORD SRL, Parco Abate 12/S, AVELLINO • KONIG SRL, Via Castaldi 28, MILANO • MA.CO. PROGRAMS SAS di Sergio Manfrè & C., Via Milano 22, CANTU' (CO) • NEWEL SRL, Via Mac Mahon 75, MILANO • PLAYER di Santuzzi, Via Balbiani 6/A, FORLI' • PRIMA INFORMATICA, Via G. Salvemini 124, MOLFETTA (BA) • RANDELLINI VALERIO, Via Campo di Marte, AREZZO • SIMPLE SOFT SAS, Via Conegliano 5, MILANO • SINCRON di Savio Paolo, Via Cassia 701, ROMA • SOFTIME di Ciceri Renato, Via Fiamminghi 43, CANTU' • SUPERGAMES SAS, Via Vitruvio 38, MILANO • TC CENTRO PAVIA SRL, Strada Statale dei Giovi, S. MARTINO SICCOMARIO (PV) • TC DATA SRL, Via Fossano 16 - CANTU' • TEKNODATA di Pincioth Enzo, Via Nazionale Adriatica 563/5, ROSETO DEGLI ABRUZZI (TE) • WARE SNC, Via dei Casceri 31, CITTA' DI CASTELLO (PG)



Lit. 59.000 + IVA

ANKO MOUSE

PAD MOUSE IN OMAGGIO!

INFORMAZIONI PER RIVENDITORI: TEL. 0341/284597

EXECUTIVE COMPUTER DEALER

Via Bovara, 16 - LECCO

Uffici e Magazzino:

Via Buozzi, 23

22053 LECCO (Co)

Tel. 0341/282614 r.a.

Fax 0341/283759

PER INFORMAZIONI:

NOME

DITTA

INDIRIZZO

TEL.

ATTIVITA'

EXECUTIVE

AMIGA

AMIGA 500	750.000
MB 114 A (drive esterno per A 500)	195.000
AMIGA 2000 (1 drive)	1.695.000
AMIGA 2000 (2 drive)	1.895.000
MONITOR COMMODORE 1084	495.000
MONITOR PHILIPS 8802	370.000
MONITOR PHILIPS 8833	470.000

DIGITALIZZATORI: PRO SOUND DESIGNER	180.000
FRAME SNAPPER	590.000
(digitalizzazione video tempo reale)	
HARDWARE E SOFTWARE: TUTTE LE NOVITÀ!	

MB 286

L'AT SUPERLATIVO

MB 286/20	2.490.000
(512 Kb, 80286 10 MHZ o wait state, 2 seriali, 1 parallela, 1 disk drive 1.2 Mb, 1 hard disk 20 Mb, tastiera estesa)	
MB 286/40	2.990.000
(come sopra ma con hard disk da 40 Mb/40ms)	
MB 286/70	3.490.000
(come sopra ma con hard disk da 70 Mb/25ms)	

COMBINAZIONI SCHEDA GRAFICA + MONITOR
DISPONIBILI PER LA SERIE MB 286

HERCULES + MONITOR MONOCROM.	390.000
(14", schermo piatto, fosfori bianchi)	
VGA + MONITOR/MONOCROMATICO	790.000
(14", schermo piatto, fosfori bianchi)	
VGA + MONITOR COLORI	1.490.000
(14", schermo piatto, pitch 031)	

SOFT center

TUTTE LE NOVITÀ
SOFTWARE PER TUTTI
I COMPUTER

ATARI

1040 STF	850.000
SM 124 (monitor monocromat.)	230.000
MEGAFILE 20 (hard disk 20 Mb)	850.000
PRO SOUND DESIGNER (digitalizz.)	135.000
MB 114 (disk drive 1 Mb)	250.000
MEGA ST 2/MEGA ST 4/LASER	P.S.R.

stair

la tua stampante

LC 10	490.000
(144 CPS, NLQ, font residenti, paper park)	
LC 10 C	590.000
(come sopra ma a colori)	
LASER PRINTER 8	P.S.R.
(Canon engine, 8 pag. min., 1 Mb RAM, 2 font)	

NOVITÀ

LC 24/10	790.000
(versione a 24 aghi della fantastica LC 10 - 160 CPS)	

AMSTRAD

PC 1640 MD con HD 20Mb	1.890.000
PC 1640 ECD con HD 20 Mb	2.490.000
AMCARD 32 (hard disk card 32 Mb)	890.000
DRIVE 3.5" per PC 1640	250.000
PPC 512/PPC 640	P.S.R.

TELEFAX

delle migliori marche

a partire da 1.695.000

MEGABYTE

VENDITA ANCHE PER CORRISPONDENZA
PREZZI IVA ESCLUSA



UFFICI: VIA CASTELLO, 1 - 25015 DESENZANO del Garda (Bs) - Tel. 030/9144880
SHOWROOM: PIAZZA MALVEZZI 14 - 25015 DESENZANO del Garda (Bs)