

VDU Driver

Un ponte fra Kernel e System Extension

di Bruno Rosati

In questa puntata parleremo del «nodo kerneliano» dell'unità di visualizzazione e i livelli d'interazione con l'ultimo sistema del Kernel (Sprite Manager) ed i (primi) moduli di estensione

Comandi della Video Display Unit

VDU	Funzione
0	Nessun effetto
1	Invia il successivo carattere solo alla stampante
2	Attiva la stampante
3	Disattiva la stampante
4	Scriva a partire dal cursore di testo
5	Scriva a partire dal cursore grafico
6	Attiva il driver dello schermo
7	Genera un «bip»
8	Muove il cursore a sinistra di una posizione
9	Muove il cursore a destra di una posizione
10	Fa scendere il cursore di una linea
11	Fa salire il cursore di una linea
12	Pulisce l'area di testo
13	Muove il cursore all'inizio della linea attuale
14	Attivazione del «modo a pagine»
15	Disattivazione del «modo a pagine»
16	Pulisce l'area grafica
17	Definisce il colore di testo
18	Definisce il colore dei grafici
19	Definisce il colore logico (setting palette)
20	Reinizializza i colori logici
21	Disattiva il driver dello schermo
22	Seleziona il modo di visualizzazione
23	Comando polifunzionale
24	Definisce una finestra grafica
25	Comando per il Plot
26	Reinizializza la finestra di default
27	Nessun effetto
28	Definisce una finestra di testo
29	Definisce l'origine grafica
30	Porta il cursore di testo all'origine
31	Muove il cursore di testo
127	Funzione di «Delete»

Figura 1

Tutto ciò che concerne la visualizzazione fa esplicito quanto ovvio riferimento alla Video Display Unit.

Le modalità di visualizzazione (Screen Modes) il tipo di «oggetto» visualizzabili (testo e/o grafica) e perfino l'elementarissima gestione del cursore, come il numero dei colori e l'importante setting della palette — Extended Colour Fill compreso —; il cicalino, gli indispensabili management del pointer e del mouse, la configurazione dello schermo e il multiwindowing system.

Come abbiamo visto due articoli fa, le informazioni vengono prodotte dal sistema per l'output dei caratteri.

Fra tutte le informazioni producibili — indirizzabili verso device quali la porta seriale, file e printer — quelle che nel caso in particolare prenderemo in considerazione sono le richieste di visualizzazione.

Tali output appena usciti dal C.O.S. viaggeranno verso il Video Controller il quale una volta letti i contenuti dei relativi blocchi di memoria, opererà la sua conversione digitale-analogica riproducendo l'informazione ricevuta in una forma che sarà finalmente in grado di «guidare» la visualizzazione a video.

VDU-commands

Se noi, ad esempio per scrivere questo articolo, digitiamo sulla tastiera le lettere che una dopo l'altra compongono le parole che abbiamo deciso di usare, altro non facciamo che inviare continui e precisi comandi al VDU device-driver. In conseguenza di ciò, possiamo tranquillamente affermare che l'intera tabella dei codici ASCII altro non è che un insieme di comandi.

Ma se è vero come è vero che una volta ricevuto l'ordine, il device VDU pilota qualsiasi carattere (ed ovviamente qualsiasi segno grafico) verso lo scher-

mo, è altrettanto vero che questi caratteri-informazione non sono tutti uguali. E non siamo certo a disquisire sul fatto che il comando ASCII della lettera «a» è diverso da quello della «A»; ciò che ci portiamo invece a distinguere, e finalmente a conoscere più da vicino, sono i primi 32 codici dello standard. Di fatto definibili quali chiamate a VDU. Caratteri speciali singoli o in sequenza che per essere riconosciuti per tali, andranno indirizzati verso la VDU attraverso l'uso della chiamata a sistema OS_WriteC.

L'intera lista dei primi 32 codici relativi è quella che potete vedere in figura 1. Leggendone attentamente le varie attribuzioni fatte, è possibile già farsi un'idea di quelle che possono essere le potenzialità del set a disposizione.

Cercando un collegamento il più possibile cronologico con quella che è la trattazione fatta dal Programmer's Reference Manual, estrapoliamo subito il comando VDU 22 (Change Display Mode) e proviamo ad amplificarne il concetto. Magari facendo un altrettanto immediato riferimento con quelli che sono i dati relativi ai contenuti dello stesso VDU 22 e gli Screen Mode archimedeanamente possibili che abbiamo inserito in figure 2 e 3.

Ebbene, come è possibile notare, per ogni chiamata del tipo:

VDU 22, <mode>

lo stesso parametro <mode> potrà variare fra i valori compresi fra 0 e 28. A ciascun valore corrisponderà il relativo screen mode.

Fra tabelle e conoscenze personali sembrerebbe tutto facile. Eppure non basta compilare una simile chiamata, scrivendo il VDU 22 e, dopo la virgola, selezionare un parametro qualsiasi. Come a molti è già noto, lo Screen Mode di Archie (come di altri personal) dipende dalle caratteristiche del monitor a disposizione. Tali capacità di risoluzione

MODI DI VISUALIZZAZIONE

Modo	Risoluzione	Colori	
0	640 x 256	2	TV-monitor e Multisync
1	320 x 256	4	TV-monitor e Multisync
2	160 x 256	16	TV-monitor e Multisync
3	Solo Testo	2	TV-monitor e Multisync
4	320 x 256	2	TV-monitor e Multisync
5	160 x 256	4	TV-monitor e Multisync
6	Solo Testo	2	TV-monitor e Multisync
7	TELETEXT	16	TV-monitor e Multisync
8	640 x 256	4	TV-monitor e Multisync
9	320 x 256	16	TV-monitor e Multisync
10	160 x 256	256	TV-monitor e Multisync
11	640 x 250	4	TV-monitor e Multisync
12	640 x 256	16	TV-monitor e Multisync
13	320 x 256	256	TV-monitor e Multisync
14	640 x 250	16	TV-monitor e Multisync
15	640 x 256	256	TV-monitor e Multisync
16	1056 x 256	16	TV-monitor e Multisync
17	1056 x 250	16	TV-monitor e Multisync
24	1056 x 256	256	TV-monitor e Multisync
18	640 x 512	2	Multisync
19	640 x 512	4	Multisync
20	640 x 512	16	Multisync
21	640 x 512	256	Multisync
25	640 x 480	2	Multisync - Monitor-VGA
26	640 x 480	4	Multisync
27	640 x 480	16	Multisync
28	640 x 480	256	Multisync - Monitor-VGA
23	1152 x 896	2	HiRes monocromatico

Figura 2

nell'Archimedes vengono catalogate in 4 differenti «type».

Il «type» zero (TV-monitor a media risoluzione); il type «uno» (il classico multisync); il type «due», ovvero monitor monocromatici ad alta risoluzione e il type «tre», quello della VGA con refresh a 60 Hz.

Per come abbiamo stampato le due tabelline comunque, la situazione apparentemente contorta dovrebbe chiarirsi rapidamente. Se con il primo raggruppamento è difatti possibile agire in relazione a monitor sia di tipo «zero» che di tipo «uno», il secondo sarà invece disponibile solo per un multisync — tra l'altro con l'evidente eccezione dei modi 25, 26, 27 e 28, possibili anche con un VGA.

Il modo 23 infine è l'unico a disposizione del tipo «due», ovvero monitor monocromatici ad alta risoluzione e frequenza di lavoro pari a 61,2 Hz.

Di comando in comando e prima di altre chiamate dalla definizione più esplicita, trovo interessante chiarire subito il significato legato alla definizione del VDU 23, quale «insieme di comandi».

Composta nella forma sintattica: VDU 23, <command>, <par1> - <par8> tale chiamata, da kernel puro, si propone al programmatore come un coman-

do *multipurpose*. Una proposta particolarmente elastica, giacché può essere parametrata con nuove differenti funzioni operative svolte dalla stessa VDU. Funzioni quali il controllo dello stato del cursore, il tipo di pattern e relativi colori dell'Extended Colour Fill, lo scroll di una pagina di testo o il Clear di un blocco delimitato della stessa text-window, la scelta del size da imporre ad un set di caratteri. Poi le chiamate «riservate» ai passaggi d'interazione con il Font Manager e lo Sprite Manager, rispettivamente: VDU 23, 25-26 e VDU 23, 27.

Che bello, già si parla d'interazione? Calma, calma. A tale livello interattivo, essendo una sorta di piano superiore, proveremo di accedervi più avanti e non prima di aver visitato l'intero piano per così dire, orizzontale.

Sempre compreso nel gruppetto multipropositivo del VDU 23 ci siamo al riguardo tenuti arbitrariamente per ultima la proposta del VDU 23,0 che, permettendo la funzione del Setlace per l'interlacciamento del segnale video, ci offre il pretesto d'introdurre il concetto della configurazione dello schermo.

Sì, come già detto, Archie è in grado di adeguare il suo output video in diversi modi e la gestione sotto RISC-OS consente di svolgere il setting in questione anche attraverso l'uso di coman-

CHANGE DISPLAY MODE (VDU 22)

Sintassi: VDU 22, <mode>

Parametri: <mode> Select dello screen mode

Uso: Per selezionare lo screen mode in relazione al tipo di monitor in uso.

TYPE	MONITOR
0	Tv-monitor
1	Multisync
2	Monitor monocromatico HiRes (61.2 Hz)
3	VGA-monitor (60Hz)

Figura 3

di particolarmente dedicati, quali:

***Configure MonitorType;** con il quale si seleziona il tipo di monitor a cui Archie, al momento dell'accensione si setterà per default. Come già detto sono 4 i tipi di monitor a disposizione ed oltre ad essere settabili attraverso il comando «programmatorio», possono essere attivati volta per volta, al momento dell'accensione del computer pigiando sul keypad il numero equivalente.

***Configure Sync;** è a sua volta predisposto alla sincronizzazione verticale — con: *configure sync 0 — oppure a quella composita: *configure sync 1.

Un comando questo «futurista», giacché adesso come adesso i computer della Acorn non possono essere sincronizzati che in composito.

***Configure TV;** è infine un VDU-command con il quale è possibile allineare ed interlacciare il video.

Sintatticamente l'operazione si riduce in tal modo:

*Configure TV [<vert align> [[,] <interlace>]] dove il parametro <vert align> permette un range di spostamento da 0 a 3 linee verso l'alto oppure di 4 linee verso il basso (in quest'ultimo caso scrivendo un valore compreso fra 255 e 252).

*Configure TV 0,1 è il valore di default con nessun spostamento ed interlace in off.

Colour, palette e pattern

Una fra le parti di maggiore importanza dell'unità di visualizzazione è quella della gestione della palette dei colori. Un vero e proprio subcontroller compreso nella struttura fisica del VIDC stesso, con il quale è possibile determinare i rapporti esistenti fra il numero di un colore immagazzinato nella memoria di schermo — ovvero il «colore logico» e l'effettiva informazione-colore indirizzata al monitor — ovvero il «colore fisico».

La VIDC-palette è programmata in modo che il controllo della intensità del segnale sia effettuato sulle tre componenti base RGB. Tre rispettive barre di controllo a 4 bit ciascuna per un totale di 4096 colori fisici visualizzabili.

Il controllo dei colori allo schermo, si estende anche sulle zone del bordo e quelle del puntatore del mouse. Per modificare il modo con il quale la palette definisce il rapporto fra colori fisici e colori logici è possibile ricorrere all'uso del comando VDU 19 (set palette) sintatticamente rappresentabile nella struttura:

```
VDU 19, <logica colour>, <mode>, <red>, <green>, <blue>
```

dove se il parametro <logica colour> è il colore da settare e che le tre componenti RGB rappresentano l'evidente livello dell'intensità da assegnare, è sul concetto di <mode> che vorrei attirare la vostra attenzione. Tale parametro difatti riveste la massima importanza.

È attraverso l'assegnazione del valore al <mode> che l'intero ordine impartito col VDU 19 può essere eseguito compiutamente.

Guardando al riguardo la tabella di figura 4, possiamo renderci più facilmente conto di quanto sia importante e in quale misura risulti determinante il valore assegnabile al <mode>.

Per default, Archie, o per meglio dire: il RISC-OS, usa un setting-mode compreso nei valori fra 0 e 15, dove il numero del colore attuale viene stabilito dal rapporto fra il colore logico e il numero di bit usati per ogni pixel.

Al riguardo la comparazione della tabella di figura 4 con quella di figura 5 ci chiarisce meglio l'intera faccenda (e se aggiungiamo un esempio del tipo: VDU 19, 5, 12, 0, 0, 0) unendo le due cose potremmo dedurre facilmente che impartendo un settaggio del colore logico di valore 5 ed effettuandolo in modo-12, con le tre componenti RGB (0, 0, 0,) completamente ignorate, il logical colour 5 verrà identificato — 7 bit per pixel — con il colore 12. Ovvero il colore blue-yellow in flashing, com'è riscontrabile dalla figura 6. Cosa vuol dire flashing colour? Semplicemente che l'effetto che si verifica in schermo nell'alternarsi di due colori differenti — con un intervallo di tempo programmato — darà luogo allo stato ottico di un terzo colore.

Ma in definitiva quanti e come sono intabellati i colori di Archie?

Il RISC-OS, a seconda dei propri Screen Mode, ripartisce i colori distribuendoli entro i limiti imposti di: 2, 4, 16 e 256.

Nei primi tre modi il procedimento è

svolto direttamente sui colori fisici corrispondenti in tabella; nel modo-256 invece ciascun pixel «colorabile» è rappresentato da un valore stabile ad 8 bit. Sei dei quali sono relativi al colore logico, mentre gli altri due vengono considerati come tinte, ovvero: le dosi di grigio che dovranno essere assommate al colore di base.

In pratica lo stato logico di una «tinta» è per molti versi simile ad un vero e proprio controller, predisposto al corretto controllo delle gradazioni di grigio. Una sorta di controllo fine.

Tale metodo è denominato a forma interna e dà il risultato che i 256 colori si ricaveranno dalla miscelazione di 64 colori di base, a ciascuno dei quali saranno associate 4 possibili gradazioni cromatiche.

Se si osserva la figura 7 è possibile chiarire ulteriormente il concetto di metodo a tinta. Il controllo su ogni singolo componente (i bit più significativi del Rosso, quelli del Green e quelli del Blue) più i due bit relativi all'ulteriore controllo sull'insieme delle tre componenti, considerato come somma.

Un altro metodo, a sua volta, permette un controllo separato delle tinte riproducibili, attraverso l'uso di una serie di comandi dedicati, predisposti, come ad esempio il VDU 17, al change del colore dei caratteri, o come il VDU 23, 17, 5: che muta foreground, background, etc. La struttura della «color table» della forma «esterna» comunque non corrisponde alla attribuzione dei bit in quella del «metodo interno». E siccome il VIDC è in grado di riconoscere solo questo ultimo, se ne deduce che le informazioni indirizzategli per mezzo del controllo separato, saranno sempre convertite in quelle corrispondenti del metodo interno.

Un altro passaggio importante nella gestione del colore sotto RISC-OS è quello che va sotto il concetto di Extended Colour Fill. I pattern; che svolgono l'interessantissima funzione di aumentare l'apparente numero dei colori con l'aggiunta di un effetto «tessitura». (Quasi quasi un Hold And Modify di amighevole memoria!)

Il comando principale dedicato al metodo di riempito esteso, è il VDU 23, con <mode> settabili da 2 a 5 per definire quattro diversi pattern colorati, così come segue:

```
VDU 23, 3 ..... set pattern 1
VDU 23, 3 ..... set pattern 2
VDU 23, 4 ..... set pattern 3
VDU 23, 5 ..... set pattern 4
```

A tali comandi-base andranno poi ad aggiungersi ben 8 parametri, in modo da realizzare una forma sintattica del genere:

```
VDU 23, <2:3:4:5>, <n1>, <n2>, <n3>, <n4>, <n5>, <n6>, <n7>, <n8>
```

dove le 8 «enne» altro non sono che i colori assegnabili ad ogni fila della tessitura del pattern. Dal colore n1 che assegna una colorazione (n1, dovrà essere sostituito dal numero del colore che vogliamo) alla fila più in alto, dove con il colore «n8» si completa il tessuto di colori.

Archie dispone di due tipi di pattern. Uno compatibile BBC/Master, l'altro tutto RISC-OS. Il primo tipo sarà di default e l'altro attivabile solo in alternativa, attraverso l'uso del comando VDU 23, 17, 4.

Mouse e pointer

Le caratteristiche fisiche e le qualità grafiche del mouse, la sfera del movimento e la serie dei bottoni, col pointer grafico a riflettere in screen i movimenti eseguiti sull'oggetto... insomma, la validità pratica del topolino magico, è ormai cosa notoria, accettata, vista e rivista.

Nel caso specifico dei VDU-driver, ci è quindi sufficiente trovare ed analizzare quali sono le modalità di controllo che il RISC-OS predispone all'uso del programmer. Qual è cioè il possibile grado d'intervento permesso sul meccanismo equivalente dei movimenti «grafici» del pointer in rapporto a quelli «fisici» del mouse. La possibilità d'intervenire per regolare la corsa a seconda delle caratteristiche del programma su cui bisognerà operare.

A tal proposito, fra i «commands» della VDU, troviamo due chiamate particolarmente dedicate. Una per il controllo della velocità del pointer, l'altra per l'on/off in screen dello stesso.

La prima è sintatticamente così composta:

```
*Configure MouseStep <n>
```

dove <n> può essere un valore compreso fra 1 e 127. Un range di velocità assolutamente esagerato, giacché usualmente, sono sufficienti i valori da 1 a 3, per identificare le tre comuni mouse-speed: slow, medium e fast. Settare già ad un valore 3 si è cioè in grado di far scorrere il pointer in maniera estremamente rapida sullo schermo. Se ad esempio, volessimo tirar su ancora la velocità portandola anche solo a n=10, ciò equivarrebbe a rendere inutilizzabile il pointer. Questo difatti schizzerebbe imprevedibile da una parte all'altra dello schermo!

L'intervento del setting-speed del pointer è cosa comunque estremamente useful e per un programmer non mancano le occasioni di andare a vivisezionare un pannello come l'application «configure».

SET PALETTE (VDU 19)

Sintassi: VDU 19, <logical colour>, <mode>, <red>, <green>, <blue>

Parametri: <logical colour> codice numerico del colore
<mode> modalità del settaggio
<red>, <green>, <blue> informazioni relative al colore fisico.

Uso: Definisce i colori della palette producendo una relazione fra il colore fisico e logico. L'azione si determina solo con la conferma del <mode> operativo.

SELECT MODE

mode= 0-15 Colore logico = Colore attuale (RED-GREEN-BLUE IGNORED)

mode= 16 Colore logico = Unità RED-GREEN-BLUE

mode= 17 Definisce il primo flash della "palette logica"

mode= 18 Definisce il secondo flash della "palette logica"

mode= 24 Definisce il colore del bordo attraverso il setting delle unità RED-GREEN-BLUE (Il colore logico non è usato)

mode= 25 Definisce il colore logico (1-3) del puntatore con le unità RED-GREEN-BLUE

Figura 4

L'altro comando a disposizione, delegato come dicevamo prima all'ON/OFF del pointer sullo schermo, si struttura in maniera molto semplice:

*Pointer [0:1]

dove lo stato parametrico «zero» equivale all'OFF mentre l'uno alla posizione attiva dell'ON. Tali condizioni hanno un rapporto diretto con la chiamata OS_Word 21,5 ed OS_Word 21,6 rispettivamente.

Sprite Manager

Scivolando sulla descrizione dei VDU-command, avevamo lasciato in sospeso l'importantissimo discorso dell'interazione fra l'unità di visualizzazione, moduli esterni e sistemi del kernel. C'erano in particolare due chiamate dedicate, pronte allo scopo. La prima — VDU 23,27 — porta al piano superiore dove ci s'introduce nell'ambiente dello Sprite Manager.

Ancora prima di porci la domanda: cos'è e in quale misura è possibile l'interazione fra VDU e Sprites? — è il caso comunque di vedere un po' più in dettaglio il significato e la valenza del concetto di Sprite.

Bene. Immaginiamo di aver eseguito un piccolo disegno (il classico pupazetto o un più serio logo) e di volerlo gestire autonomamente da tutta la pagina (grafica) avendo l'opportunità di

farlo spostare da un posto all'altro dello screen e manipolarlo in tutta la sua interezza. (Senza che il pur minimo pixel sfugga al nostro controllo!).

In pratica ciò che serve è la formidabile possibilità di poter gestire in proprio, blocchi separati di screen memory che contengano, in un array di pixel, la forma grafica da noi delimitata. Lo «spirito» archimedeo che il RISC-OS dispone all'utenza è quanto di più completo possa esser messo a disposizione del programmatore.

I suoi attributi standard comprendono: il *Name*, usato per identificare lo sprite stesso; il *Number* dello screen mode di cui lo sprite, così come usa il manuale, «imita» il formato; le dimensioni in altezza e larghezza e poi altre due informazioni opzionali: l'ovvia palette di colori usata dallo sprite e la maschera trasparente (ovvero l'effetto di scontornamento che permette di perimetrare lo sprite lasciando trasparire il

fondo su cui poggerà).

Lo sprite può essere definito grabbando porzioni di schermo oppure plottando con il predisposto VDU 25 che può operare direttamente sull'oggetto ed a prescindere dalla pagina grafica di lavoro. Allo stesso livello di manipolabilità il RISC-OS consente di plottare lo sprite scalandolo a qualsiasi size e di alterarne la colorazione.

Fra SWI-call e comandi dedicati equivalenti, passiamo ad analizzare la VDU 23,27 che nella sua forma estesa è sintatticamente organizzata come segue:

VDU 23,27, m,n:

dove m=0 equivale allo Sprite-command *SChoose *name* ed m=1 all'altro Sprite-command *SGet *name* limitati rispetto al VDU dedicato che può disporre anche di una serie di variabili.

All'uopo torna utile l'esempio che l'User Guide fa in riferimento alla possibilità di rappresentare in screen più for-

TABELLA RELAZIONALE INTERNA (COLORE LOGICO)

Colore Logico	Bit 1	(x) Pixel 2	Screen Mode 4
0	0	0	0
1	7	1	1
2		3	2
3		7	3
4			4
5			5
6			6
7			7
8			8
9			9
10			10
11			11
12			12
13			13
14			14
15			15

Figura 5

TAVOLE DEI COLORI FISICI

(MODO a 16 colori)

Colore Logico	Colore Fisico
0	Nero
1	Rosso
2	Verde
3	Giallo
4	Blu
5	Magenta
6	Ciano
7	Bianco
8	Flash Nero/Bianco
9	Flash Rosso/Ciano
10	Flash Verde/Magenta
11	Flash Giallo/Blu
12	Flash Blu/Giallo
13	Flash Magenta/Verde
14	Flash Ciano/Rosso
15	Flash Bianco/Nero

(MODO a 2 colori)

Colore Logico	Colore Fisico
0	Nero
1	Bianco

(MODO a 4 colori)

Colore Logico	Colore Fisico
0	Nero
1	Rosso
2	Giallo
3	Bianco

Figura 6

STATO DI DEFAULT DELLA PALETTE (Internal Setting)

Bit	Significato
0	Bit 0 della Tinta (RGB/bit 1)
1	Bit 1 della Tinta (RGB/bit1)
2	Bit 2 della componente RED
3	Bit 2 della componente BLUE
4	Bit 3 della componente RED (HIGH)
5	Bit 2 della componente GREEN
6	Bit 2 della componente GREEN (HIGH)
7	Bit 3 della componente BLUE (HIGH)

Figura 7

STATO DEI COLORI LOGICI DEI PIXEL IN UNA "PATTERN-ROW"

Bit x Pixel	Numero Colori	Pixel x Linea	Colori disponibili sotto RISC-OS
1	2	8	0,1,2,3,4,5,6,7
2	4	4	10,32,54,76
4	16	2	3210,7654
8	256	1	76543210

Figura 8

me contemporaneamente.

Un semplice esempio in Basic caratterizzato da quattro semplici linee quali:

```
FOR sprite_num% = 1 TO 8
VDU 23,27, sprite_num%, 0,0,0,0,0
PLOT &ED, x%, y%
NEXT
```

Selezionare e Plot-are sprite con il metodo dei VDU appena visti è comune una tecnica poco usata; usualmente tali comandi vengono usati solo per l'accesso all'area di utilizzo.

Font Manager

Analogamente allo Sprite Manager, anche per la gestione Font esiste la possibilità di accesso attraverso l'uso dei VDU-commands. Per l'esattezza, si tratta di due forme sintattiche quali la VDU 23,25 (in forma generica) e la VDU 23,26.

Mentre quest'ultimo comando è riservato alla selezione di un carattere (predefinito in size e risoluzione) la forma generica del primo — VDU 23,25 si predispone all'uso dei caratteri con effetto a scalino — è ripartibile in ulteriori «command» dalle funzioni particolarmente specializzate.

Avremo così a disposizione il:

VDU 23,25, 1-4,

predisposto per la funzione di trasferimento dei 16 livelli di grigio da usarsi per l'eliminazione dell'effetto di aliasing (perlappunto l'effetto scalino). Quindi il: VDU 23,25, 128-143

capace di definire i registri relativi alla palette per l'anti-aliasing nei modi a 2, 4 e 16 colori; ed infine il:

VDU 23,25, &FF

con il quale si stabilisce la definizione di uno dei 16 colori logici, sempre per l'anti-aliasing, nei modi a 256 colori.

Ebbene, tutti questi comandi sono predisposti all'interazione con il Font Manager, per la gestione congiunta — dalla richiesta, il setting e la visualizzazione — dei caratteri in screen.

Cercare un font-file per la lettura, predisporre il cachememory dei dati relativi al font-file letto, provvederne una

eventuale manipolazione (misure e size), invocarne il trasferimento alla memoria della VDU e cambiarne eventualmente i colori.

Sono queste le fasi in cui si articola la gestione del sistema dei caratteri archimediano.

A livello di VDU-command l'esempio ideale del livello d'interazione ci viene proprio dalla VDU 23,26 che nella sua forma sintattica completa è costituita come segue:

```
VDU 23,26 <font handle>, <pont size>,
<x dpi>, <y dpi>, <x scale>, <y scale>,
0,0, <font name>
```

dove è possibile leggere con una certa facilità tutti i dati relativi al tipo di set che si desidera e che il Font Manager provvederà a produrre.

Tale comando VDU ha un suo equivalente nella SWI-call Font_FindFont che lavorando in entrata — ovvero appena ricevuta l'informazione completa sulle caratteristiche del set — predisporrà allo scopo cinque registri di ricerca e manipolazione. R1 con puntatore verso il font-name; R2 e R3 per il size chiamato da x ed y; infine R4 ed R5 per la risoluzione in dot per inch sempre sui due assi x ed y.

Ad un passo dal concetto che segue (quello cioè di stampa) non ci sembra affatto scorretto, introdurci nell'insieme dei moduli di estensione, attraverso tale chiave di lettura.

Anzi, la cosa mi sembra particolarmente indicativa se consideriamo quali sono le caratteristiche del modulo in questione.

Se per font, intendiamo difatti un generico set di caratteri composti tutti con un certo stile comune, il management svolto dal Font Manager del RISC-OS a sua volta è possibile intenderlo come un sistema in cui viene concentrata tutta una serie di facility, predisposte al fine di ottenere, in output, qualsiasi stile e taglia del set di caratteri desiderato. Una delle caratteristiche di spicco del Font Manager è la sua funzione di scalabilità dei font. A prescindere dal fatto che l'archivio a disposizione contenga o meno il set di

caratteri da noi prescelto, il Font Manager ne esegue la scala richiesta per mezzo del sistema di misurazione interno ad esso. Il *measurement system* è una sorta di simulazione software della VDU che invece di analizzare e quindi manipolare il testo in termini di pixel, lo fa direttamente sulle misure in punti. L'intero procedimento è possibile grazie all'unità di misura del sistema stesso, basata su valori da un millesimo di punto (tipografico).

Eh sì; il RISC-OS pensa veramente a tutto. Anche al controllo dello store dei set, effettuando con un sistema dedicato che si basa sulla tecnica del caching.

Allo stesso modo dei VDU-driver, anche nei set del Font Manager, oltre che alla serie dei caratteri stampabili c'è il gruppo dei «non-printable character». Da intendersi come font-driver, tali caratteri speciali sono predisposti al cambio dei colori, l'alterazione della posizione di scrittura, etc.

Concludendo

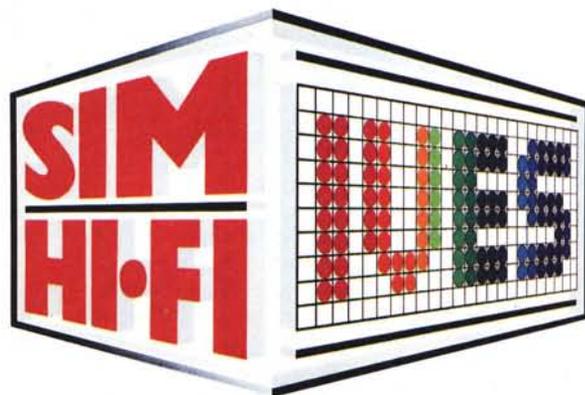
Malgrado il ritmo che si è tenuto (portandolo volutamente alto per esporre il lambiccoso sistema di visualizzazione nel modo più esaustivo possibile) il tema VDU-driver non può dirsi concluso.

Altri moduli di estensione come il Colour Trans, il Draw Module e il Window Manager, interagiscono con la Video Display Unit. Tre argomentazioni che è giocoforza rinviare a dopo l'estate con lo sfruttamento dell'area già sperimentata della «Piccola Guida Tecnica». Fra questa e lo spazio dedicato al Corso di RISC-OS per quella che sarà l'ultima overview, arriveremo a tirare finalmente le fila, e fatto il punto sulla lunga galoppata, ad organizzare la futura rubrica archimediana su nuove e spero, più argomentabili strutture.

Per questa volta è tutto. Non ci resta che farci reciproci auguri di buone vacanze. A settembre.

MUSICA PER I VOSTRI OCCHI

SOLUTIONS



ASSOCIATO
ASSOCIATI

**24° Salone Internazionale della Musica e High Fidelity
International Video and Consumer Electronics Show**

Fiera Milano • 20/24 Settembre 1990

ALTA FEDELITÀ • CAR ALARM SYSTEMS • ELETTRONICA
DI CONSUMO • HI-FI CAR • HOME VIDEO • PERSONAL COMPUTER
STRUMENTI MUSICALI • TV • VIDEOREGISTRAZIONE

**HOME
VIDEO**
5ª Rassegna delle
videocassette registrate

Ingressi: Piazza Carlo Magno - Via Gattamelata
Orario: 9.00/18.00 - Pad. 7-12-13-14-14 A/B-15
Aperto al pubblico: 20-21-22-23 settembre
Giornata professionale: lunedì 24 settembre

**VIVA
i giovani
90**
Festa per i giovani
musicisti

Segreteria Generale SIM-HI•FIVES: Via Domenichino, 11 - 20149 Milano - Tel. (02) 4815541 - Telex 313627 - Fax (02) 4980330

 **ASOEXPO**

 **FIERA
MILANO**