

Il Mouse

seconda parte

Dopo aver parlato nella scorsa puntata della funzione di un mouse in un personal computer e dopo aver visto in quale «mondo» (lo schermo video nelle varie risoluzioni) esso si muova, iniziamo da questa puntata l'analisi delle funzioni che abbiamo a disposizione per gestire correttamente il mouse da programma.

Ricordiamo che faremo particolare riferimento all'Assembler, al Pascal ed al Basic, cercando di avere sempre sott'occhio un esempio pratico da sperimentare effettivamente sul proprio personal: per la sua universalità, il mouse è gestibile via software qualunque sia l'hardware al quale è connesso, cioè qualsiasi sia il microprocessore all'interno del personal, qualunque sia la quantità di memoria presente, ecc.

Ovviamente il mouse deve essere connesso ad una porta seriale, in genere COM1 e per il resto basta scrivere programmi per poterlo gestire

Le prime funzioni

Facciamo riferimento alla figura 1, che rappresenta un riassunto conciso delle funzioni implementate in genere in un qualunque «MOUSE.COM» fornito insieme al mouse: in essa le funzioni sono ordinate per numero (valore da porre nel registro AX) e solo in prima analisi tale ordine rispecchia l'effettivo ordine con cui nel programma devono essere eseguite.

Alcune di esse in effetti si usano molto di rado, mentre altre sono molto importanti ed usate anche più volte in un programma. Nell'analizzarle dunque, più che l'ordine numerico, seguiremo un criterio più logico, andando a vedere innanzitutto quelle fondamentali.

La funzione 0: "Mouse Reset"

Si tratta in effetti della prima funzione che deve essere chiamata proprio per attivare il mouse in un certo programma: come si vede dalla figura 2, basta porre 0 in AX e chiamare l'INT 33H, ovviamente dopo aver attivato il programma «MOUSE» (o chi per lui) il quale rimane residente in memoria.

Se tutto va bene (MOUSE.COM lanciato, mouse correttamente connesso alla porta seriale e mouse che risponde esattamente alle attese del driver e cioè inviando byte ad intervalli predeterminati) allora AX ritornerà con un valore pari a -1 (0FFFFH), mentre in caso contrario in AX ci sarà (il più delle volte) un valore nullo: diciamo «il più delle volte» in quanto nel caso in cui non sia stato lanciato MOUSE.COM, allora il comportamento di INT 33H non è predicibile così come il valore che ritornerà in AX.

Sapendo che il valore ottimale è -1 si può vedere che c'è una probabilità su 65536 che viceversa torni un valore -1 come valore imprevedibile e casuale, nel caso in cui INT 33H non possa essere eseguito.

Bando alle sottigliezze, l'altro parametro fornito in uscita è contenuto in BX ed indica il numero di pulsanti presenti nel mouse: nel caso che ci siano 2 o 3 pulsanti il valore fornito è correttamente 2 o 3, mentre in generale è possibile far vedere al software solo 2 dei 3 pulsanti (in genere tenendo premuti uno o più pulsanti all'atto dell'accensione del

computer), nel qual caso la funzione fornirà un valore pari a 2, rendendo il software «sordo» ad ulteriori pressioni del pulsante centrale. Bisogna poi ricordare che «noi» sappiamo quanti sono i pulsanti, mentre il software no: in generale succede che i programmi, proprio per essere compatibili con il maggior numero di mouse in commercio, prevedano l'utilizzazione di solo 2 pulsanti, anche se viceversa la funzione in esame fornisce un valore pari a 3.

Oltre a fornire questi due valori, la funzione 0 serve ad effettuare una serie di inizializzazioni di variabili interne, che servono successivamente per gestire il funzionamento del mouse stesso; in pratica (alcuni termini verranno chiariti nel seguito), i settaggi iniziali sono i seguenti:

- la posizione del mouse è fissata al centro dello schermo;
- il «cursor flag» viene posto a -1, disabilitando la visione sullo schermo di un cursore connesso al moto del mouse;
- viene settato come cursore grafico un freccia, mentre come cursore in modo «testo» viene prefissato il solito cursore pieno ed «invertente», in modo da essere sempre visibile qualunque siano i colori dello sfondo che attraversa;
- viene disattivata la maschera di attivazione dell'interrupt, che perciò se necessario dovrà essere settata da programma;
- viene abilitata l'emulazione della penna ottica;
- viene settato il rapporto, tanto orizzontale quanto verticale, tra lo spostamento nelle due direzioni (espresso in una curiosa unità di misura che gli americani hanno pensato bene di chiamare «mickey») ed il numero di pixel di spostamento sullo schermo;
- vengono settati i valori minimi e massimi orizzontali e verticali per le coordinate del cursore sullo schermo, in funzione della scheda grafica adottata.

Le funzioni 1 e 2: "Cursor Enable" e "Cursor Disable"

Il fatto che il cursore sia visibile o meno sullo schermo (a parte quando viene portato ai bordi per cui può parzialmente se non totalmente scompa-

re) dipende da una particolare variabile interna detta «mouse flag»: se tale variabile vale 0 allora il cursore viene mostrato in corrispondenza delle coordinate correnti, mentre un valore diverso da 0 lo farà scomparire.

La funzione 1 (si veda la figura 3: non dà parametri in uscita) non fa altro che incrementare di 1 la variabile di cui sopra: se raggiunge un valore pari a 0, allora e solo allora il cursore viene mostrato.

Invece la funzione 2 (si veda la figura 4: anche questa dà parametri in uscita), oltre a decrementare il «mouse flag», in contrapposizione alla funzione 1, va a disabilitare comunque il cursore dallo schermo.

In effetti il tutto è alquanto contorto: il «mouse flag» è in realtà un contatore (posto al reset pari a -1, come detto) che a seconda del suo valore permette o meno la visione del cursore, in stretta dipendenza da quale delle due funzioni (1 o 2) ne ha mutato il valore.

Ecco che solo dopo il reset (flag = -1) la funzione 1 fa mostrare il cursore (il flag diventa 0), ma successive chiamate alla stessa funzione 1 non fanno altro che incrementare tale «flag» inutilmente (dato che il cursore era già acceso!).

Ogni volta poi che viene chiamata la funzione 2 il cursore viene spento, indipendentemente dal valore del «flag» che comunque viene decrementato.

Come dire che bisogna tener sempre in conto di quante volte viene chiamata la funzione 1 e quante la 2 per non avere sorprese: chiamando due volte la «1» il flag varrà 1 (cursore on), e chiamando successivamente una volta sola la «2» (il cursore va off), il flag varrà 0 ed una successiva chiamata alla «1» NON accenderà il cursore, dato che ora il flag varrà 1, fatto che non consente l'accensione sullo schermo del sospirato cursore.

Peggio è se si chiama la funzione «2» parecchie volte: ci vorranno altrettante chiamate alla «1» prima di poter vedere effettivamente il benedetto cursore...

In sintesi dunque bisogna sempre contare quante volte viene chiamata l'una o l'altra funzione: perché sia stato prescelto un funzionamento così bizzarro non si sa...

Tra l'altro il nome stesso della funzione 1 può trarre in inganno in quanto ci si aspetterebbe ad ogni sua chiamata l'apparizione del cursore sullo schermo: perché complicare inutilmente una cosa semplicissima?!

Le funzioni 7 ed 8: "Set min/max horizontal/ vertical position"

In questo caso le funzioni in esame non sono state complicate inutilmente! Infatti il loro compito è quello di definire una finestra entro cui si può muovere il

cursore, prestabilendone le coordinate minime e massime tanto orizzontali quanto verticali. Non è obbligatorio chiamare queste due funzioni in quanto abbiamo già visto che per default al reset del mouse la finestra di visibilità viene posta coincidente con tutto lo schermo video: invece nel caso in cui si voglia restringere il campo di azione, ecco che con la funzione 7 definiremo il valore minimo e quello massimo che possono assumere le coordinate orizzontali, mentre la funzione 8 setta le omologhe quantità per le coordinate verticali.

Bisogna dunque fare attenzione che non si tratta di definire una «window», per cui bisogna fornire le coordinate dei vertici diagonalmente opposti (di solito quello in alto a sinistra e quello in basso a destra), anche perché una volta definiti i valori minimi e massimi, le coordinate del cursore all'interno della zona rettangolare non vengono minimamente toccate: se ad esempio si decide di restringere il campo di azione per le «x» tra i valori 100 e 400, allora le coordinate del cursore all'interno della zona prescelta varieranno sempre tra 100 e 400 (sono cioè sempre coordinate «assolute») e non, per intenderci, tra 0 e 300 e cioè «relative».

Ovviamente e generalmente il programma MOUSE.COM, e perciò le routine che implementano tali funzioni in esame, è protetto da situazioni in cui si diano dei valori «minimi» maggiori di quelli «massimi», come pure se si danno valori maggiori di quelli consentiti e legati alla risoluzione della scheda video.

La funzione 3: "Get Mouse Position and Cursor Status"

Si tratta senza dubbio di una funzione molto importante (si veda la figura 7), in quanto consente, ad ogni sua chiamata, di conoscere la posizione del cursore sullo schermo, si badi bene, anche quando il cursore non è materialmente visibile sullo schermo in quanto spento dalla funzione 2.

In particolare la funzione 3 fornisce, ad ogni chiamata, tre valori posti nei registri BX, CX e DX e rispettivamente lo stato dei pulsanti, la posizione orizzontale e quella verticale del cursore.

AX = 0H	Mouse Reset
AX = 1H	Cursor Enable
AX = 2H	Cursor Disable
AX = 3H	Get Mouse Position and Button Status
AX = 4H	Set Mouse Position
AX = 5H	Get Button Press Information
AX = 6H	Get Button Release Information
AX = 7H	Set min & max horizontal position
AX = 8H	Set min & max vertical position
AX = 9H	Set Graphic Cursor Block
AX = 0AH	Set Text Cursor
AX = 0BH	Read Motion Counters
AX = 0CH	Set User-defined subroutine
AX = 0DH	Enable Light Pen Emulation
AX = 0EH	Disable Light Pen Emulation
AX = 0FH	Set Mickev/Pixel Ratio
AX = 10H	Window Conditional Off
AX = 12H	Set Large Graphic Cursor
AX = 13H	Set Speed Threshold

◀ Figura 1 - Riassunto delle funzioni per gestire un mouse tramite l'INT 33H: a parte la funzione 12H, sono tutte funzioni standard che devono essere comunque implementate nel file MOUSE.COM che accompagna il mouse. Occasionalmente ne vengono create di nuove, fuori standard, come appunto la 12H.

Figura 2 - La funzione 0 non ha parametri in input, ma fornisce due valori in uscita. ▼

AX = 0H	Mouse Reset
INPUT	-
OUTPUT	AX = mouse status BX = number of buttons

AX = 1H	Cursor Enable
INPUT	-
OUTPUT	-

Figura 3 - La funzione 1 viceversa non ha né parametri in input, né valori in uscita.

AX = 2H	Cursor Disable
INPUT	-
OUTPUT	-

Figura 4 - Analogamente alla precedente, la funzione 2 non ha parametri in input, né in output.

AX = 7H	Set min & max horizontal position
INPUT	CX = min horizontal position DX = max horizontal position
OUTPUT	-

Figura 5 - La funzione 7 riceve in input due valori che rappresentano il minimo e massimo valore per le coordinate orizzontali.

AX = 8H	Set min & max vertical position
INPUT	CX = min vertical position DX = max vertical position
OUTPUT	-

Figura 6 - La funzione 8 riceve in input due valori che rappresentano il minimo e massimo valore per le coordinate verticali.

AX = 3H	Get Mouse Position and Button Status
INPUT	-
OUTPUT	BX = button status CX = cursor x position DX = cursor y position

Figura 7 - Questa è una funzione molto utile ed importante in quanto consente di leggere la posizione corrente del mouse nello schermo video e lo stato dei due o tre pulsanti.

Mentre queste ultime due sono espresse al solito in pixel (ad esempio in una CGA varieranno, le «x», tra 0 e 639, mentre le «y» tra 0 e 199, sempreché non siano state usate le funzioni viste in precedenza per restringere il campo d'azione del mouse stesso), lo stato dei pulsanti è codificato nei tre bit meno significativi di BX, intendendo che un «1» significa pulsante premuto.

In particolare si ha:

- il bit 0 (il meno significativo) rappresenta lo stato del pulsante di sinistra;
- il bit 1 indica lo stato del pulsante di destra;
- il bit 2 indica infine lo stato del pulsante centrale.

Va da sé che il mouse ha solo due pulsanti (quello originario della Micro-

soft) oppure se ne ha tre, ma ne è stato predisposto il funzionamento con solo due, allora solo i bit 0 ed 1 potranno variare.

Un esempio

In figura 8 vediamo un esempio scritto in Pascal, ma facilmente convertibile in Basic secondo quanto detto la scorsa puntata; si tratta di un programmino di test del mouse, che effettua le seguenti operazioni:

- entra in grafica alla massima risoluzione permessa dalla scheda video;
- attiva il mouse (ricordarsi di lanciare MOUSE.COM prima di entrare in Turbo Pascal!!!) con la funzione 0;
- setta una finestra di lavoro con le

```

uses graph,crt,dos;
var gd,gm : integer;
    reg : registers;
    CX,DX : integer;

procedure mouse(ax : word);
begin
  reg.AX := ax;
  intr($33,reg);
end;

function inttostr(x : integer) : string;
var s : string;
begin
  str(x,s);
  inttostr := s;
end;

begin
  gd := detect;
  initgraph(gd,gm,'');
  CX := getmaxx;
  DX := getmaxy;
  mouse(0);                               {mouse reset}
  reg.CX := 10;
  reg.DX := getmaxx - 10;
  mouse(7);                               {set hor. position}
  reg.CX := 10;
  reg.DX := getmaxy - 10;
  mouse(8);                               {set vert. pos.}
  rectangle(10,10,getmaxx - 10,getmaxy - 10);
  mouse(1);                               {cursor on}
  repeat
    mouse(3);                             {get cursor pos.}
    if CX <> reg.CX
    then
      begin
        setcolor(0);
        outtextxy(0,0,inttostr(CX));
        setcolor(getmaxcolor);
        outtextxy(0,0,inttostr(reg.CX));
        CX := reg.CX;
      end;
    if DX <> reg.DX
    then
      begin
        setcolor(0);
        outtextxy(30,0,inttostr(DX));
        setcolor(getmaxcolor);
        outtextxy(30,0,inttostr(reg.DX));
        DX := reg.DX;
      end;
  until reg.BX > 0;
  mouse(2);
  closegraph;
end.

```

Figura 8 - Piccolo esempio di come si gestisce un mouse: prima di eseguire il programma e prima ancora di entrare in Turbo Pascal, lanciare MOUSE.COM per installare l'INT 33H.

funzioni 7 ed 8;

- visualizza tale finestra con un rettangolo;

- accende il cursore sul video, una freccetta, con la funzione 1;

- entra in un ciclo nel quale si può muovere il mouse mostrando istante per istante le sue coordinate (lette con la funzione 3) in alto a sinistra nello schermo;

- testa se un qualunque pulsante è stato premuto (BX > 0): in caso negativo rimane nel loop mentre in caso positivo...

- spegne il cursore sul video con la funzione 2;

- esce dalla grafica e termina.

Con questo abbiamo finito: a risentirci nella prossima puntata.

Nuovo Word 5.

Da oggi tutto è più facile.

Parola di Microsoft.

I programmi Microsoft per word processing offrono oggi soluzioni avanzate già valide per il futuro.

Soluzioni capaci di sviluppare al massimo il tuo attuale livello di produttività, unitamente alla migliore qualità nella presentazione dei documenti.

Word 5, ultimo nato dalla famiglia Microsoft, dice l'ultima parola nel word processing e ti porta con potenza e velocità mai viste prima al più alto livello di produttività individuale e collettiva oggi possibile operando in ambiente MS-DOS e MS® OS/2.

E con risultati paragonabili a capolavori tipografici. Chi già conosce la versione precedente, può apprezzare ancora meglio i nuovi orizzonti posti da Word 5.

La sua bellezza più appariscente è la minima quantità di operazioni richieste e la loro estrema semplicità di esecuzione per ottenere documenti di qualità decisamente professionale.

Con il comodo aiuto dell'anteprima di stampa, ora integrata nel programma, puoi seguire passo dopo passo il tuo lavoro e vedere chiaramente a video come verrà stampato secondo il più avanzato concetto di WYSIWYG (ciò che vedi è ciò che ottieni).

Con estrema semplicità e velocità puoi integrare testi e tabelle, inserire grafici e figure, dati e calcoli, impostare margini, modificare colonne e formattazione. E mille altre cose cose ancora. Come dire: i

tui progetti saranno più attraenti, le tue proposte più convincenti e anche i tuoi memo diventeranno... più memorabili.

Word 5 facilita e velocizza al massimo i tempi d'impostazione di nuovi documenti sfruttando in pieno i glossari. Questi ultimi comprendono testi preconfezionati per gli usi più comuni e fogli stile contenenti formati standard e speciali: così, e anche con l'aiuto delle comode macro, puoi crearti una completa biblioteca di modelli di riferimento.

Microsoft Word 5 ha una spiccata predisposizione al lavoro di gruppo. Nella versione per rete, consente di dialogare con altri utenti per fare circolare bene e in fretta idee e progetti

fra più persone interessate e accogliere automaticamente revisioni, aggiunte, spostamenti, note e commenti che resteranno memorizzati nel documento e disponibili a richiesta in qualsiasi momento. Rivolgiti al rivenditore Microsoft più vicino: oltre a dimostrazioni e "prove su strada" ti aiuterà a scegliere l'opportunità commerciale più interessante per le tue necessità.



Per maggiori informazioni scrivi o telefona a
 Microsoft S.p.A. Milano Oltre-palazzo Tiepolo
 Via Cassanese 224, 20090 Segrate,
 tel. 02-2107201

Microsoft

Il software del tuo successo.