Text Mode Cursor

Contents [hide]

Moving the Cursor with the BIOS

Moving the cursor with the BIOS is done through Int 0x10 (The general interrupt for screen functions) with AH set to 0x02. These are the registers used:

- AH = 0x02
- BH = Display Page (This is usually, if not always, 0)
- DH = The row
- DL = The column

Then, with a quick call to interrupt 0x10, you should have yourself a movable type cursor.

Moving the Cursor without the BIOS

Without access to BIOS calls and functions, moving the cursor requires using video hardware control. Lucky it is a simple procedure.

Note, this quick example assumes 80x25 screen mode. Also note that the base port (here assumed to be 0x3D4) should be read from the BIOS data area.

Source in C

Navigation

Main Page
Forums
FAQ
OS Projects
Random page

About

This site
Joining
Editing help
Recent changes

Toolbox

What links here
Related changes
Special pages
Printable version
Permanent link

```
/* void update_cursor(int row, int col)
 * by Dark Fiber
 */
void update_cursor(int row, int col)
{
   unsigned short position=(row*80) + col;

   // cursor LOW port to vga INDEX register
   outb(0x3D4, 0x0F);
   outb(0x3D5, (unsigned char)(position&0xFF));
   // cursor HIGH port to vga INDEX register
   outb(0x3D4, 0x0E);
   outb(0x3D5, (unsigned char )((position>>8)&0xFF));
}
```

Note that the 2 parameters 'row' & 'col' passed to the function above start from zero, not from 1. And keep in mind that in/out to VGA Hardware is a slow operation. So using the hardware registers to remember of the current character location (row, col) is bad practice -- and updating position after each displayed character is poor practice (updating it only when a line/string is complete is wiser and hiding it until a user prompt is wisest)

Source in assembly

Since BIOS services can't be accessed in 64bit long mode, the following routine shows how to move cursor without BIOS in VGA text 80x25 (can be altered a bit to fit protected mode):

```
; Set cursor position (text mode 80x25)
; @param BL The row on screen, starts from 0
; @param BH The column on screen, starts from 0
;=============================================================================
set_cursor:     pushfq
                push rax
                push rbx
                push rcx
                push rdx

                ;unsigned short position = (row*80) + col;
                ;AX will contain 'position'
                mov ax,bx
                and ax,0ffh             ;set AX to 'row'
                mov cl,80
                mul cl                  ;row*80

                mov cx,bx
                shr cx,8                ;set CX to 'col'
                add ax,cx               ;+ col
                mov cx,ax               ;store 'position' in CX

                ;cursor LOW port to vga INDEX register
                mov al,0fh
                mov dx,3d4h             ;VGA port 3D4h
                out dx,al

                mov ax,cx               ;restore 'postion' back to AX
                mov dx,3d5h             ;VGA port 3D5h
                out dx,al               ;send to VGA hardware

                ;cursor HIGH port to vga INDEX register
                mov al,0eh
                mov dx,3d4h             ;VGA port 3D4h
                out dx,al

                mov ax,cx               ;restore 'position' back to AX
                shr ax,8                ;get high byte in 'position'
                mov dx,3d5h             ;VGA port 3D5h
                out dx,al               ;send to VGA hardware

                pop rdx
                pop rcx
                pop rbx
                pop rax
                popfq
                ret
```

See Also

- VGA Hardware

External Links

- http://www.bookcase.com/library/dos/ints/int10.html (dead link)
- https://web.archive.org/web/20120324083032/http://www.arl.wustl.edu/~lockwood/class/cs306/books/artofasm/Chapter_13/CH13-

2.html

Categories: Video | VGA

This page w as last modified on 30 March 2015, at 09:52.

This page has been accessed 57,555 times.