The window manager moves the mouse; applications choose the cursor

devblogs.microsoft.com/oldnewthing/20061117-03

November 17, 2006

Raymond Chen

You can sometimes narrow down the source of a problem just by looking at the screen and moving the mouse. When you move the mouse, the cursor on the screen moves to match. This work is done in the window manager in kernel mode. The mouse hardware notifies the window manager, "Hey, I moved left twenty units." The window manager takes this value, accelerates or decelerates it according to your mouse acceleration settings, calls any low-level mouse hooks that are installed, and then tells the display driver, "Move that sprite left about thirty pixels" (say). It then sets the "the mouse moved" flag so that the program who owns the window under the new mouse position will get a `WM_MOUSEMOVE` message. The window manager also sets the cursor to the "virtual cursor state" corresponding to the window beneath the cursor. The "virtual cursor state" remembers the cursor that the thread (or threads, if input has been attached) responsible for the window most recently set. Maintaining the virtual cursor state is important, for if a thread calls `SetCursor` to change the cursor to an hourglass and then stops processing messages (because it is busy), you really want the cursor to change back to an hourglass when it moves over the thread's windows. What does it mean if the cursor doesn't move at all when you move the mouse? Could it be caused by an application? If you read through the flowchart I described above, the only place applications get involved in the "move the mouse cursor" code flow is if they are filtering out the mouse motion in a low-level mouse hook. (Another way an application can "lock up" the mouse is by calling the `ClipCursor` function, but vanishingly few applications do this. I'm assuming you aren't the victim of malicious software but instead are trying to figure out what program, if any, is **accidentally** freezing the mouse.) Low-level mouse hooks are comparatively uncommon since they exact a high performance penalty on the system. If you're moving your mouse and don't see the cursor move around on the screen, my guess is that there is a problem in the kernel-mode side of the equation. If you're seeing the entire system freeze up, then it's probably a device driver that has started acting up and held a lock for too long. A flaky hard drive can have the same effect. If the window manager itself takes a page fault, it has to wait for the hard drive to page in the data. and if the window manager happened to be holding a lock when this happened, that lock is held across the entire I/O operation. If your hard drive is flaky and, say, takes ten seconds to produce a sector of data instead of several milliseconds, then it will look like the system has frozen for ten seconds, since the window manager is stuck waiting on your disk, which is in turn grunting and recalibrating in a desperate attempt to produce the data the memory manager requested.

In other words: If the cursor won’t move, it’s likely a driver or hardware problem. (Figuring out which driver/hardware will require hooking up a kernel debugger and poking around. Not for the faint of heart.)

Raymond Chen

Follow