

Analysis of KVM Switch Problems

Source: <http://linux.derkeiler.com/Newsgroups/alt.os.linux.suse/2004-06/4844.html>

From: Mark S Bilk (mark_at_cosmicpenguin.com)

Date: 06/27/04

Date: 27 Jun 2004 05:56:02 -0700

The mouse and keyboard each send digital messages to the motherboard over one or more signal wires. The messages are sent by changing the voltage on the wires back and forth between two distinct voltage levels; let's assume that these are the TTL standard levels of 0 and +3 volts (the actual values don't matter).

When you aren't moving the mouse, and aren't typing on the keyboard, the voltage levels on the wires stay quiescent — unchanging — at either 0 or +3. In order to avoid sending spurious voltage changes to the motherboard on these wires (which the motherboard controller chips will interpret as mouse/button movements or key presses), the KVM switch must maintain the required quiescent voltage levels on all the signal wires when you operate it to switch the mouse and keyboard between the two computers.

If it does so, then there is no way that changing the KVM switch setting can possibly have any effect on either computer, because what the computer sees are the same quiescent voltage levels that it would see if there were no KVM switch and you were simply not touching the mouse or keyboard.

The situation would be more complex if the mouse or keyboard sends a clock signal to the motherboard (i.e., rapidly changes the voltage on one of the wires between 0 and +3 to synchronize the communication), or if the motherboard sends such signals to the input devices. In this case, the clock signals would have to be maintained by the KVM switch during the switchover operation so that any device receiving one would continue to do so without interruption. Again, if the KVM switch does this, neither computer would be affected by a switchover. Neither computer could tell the difference between a switchover and an interval of time during which the user had simply not operated the mouse or keyboard.

So, if operating the KVM switch disturbs one of the computers, then the switch is not functioning correctly, either because it's

broken (or maybe dirty), or because it wasn't designed properly in the first place.

When a switch is operated, the metal contacts always bounce, causing very rapid multiple interruptions of the voltage being sent through the switch. If the voltage is being interpreted as a digital message (as in this case), the circuit receiving the voltage will interpret the bounce interruptions as some kind of spurious and erroneous message. Therefore, mechanical switches used in digital equipment should always be connected into electronic "debouncing" circuits, which filter out the interruptions and deliver a single transition from, e.g., 0 to +3 volts.

>From this analysis it appears that for a KVM switch to avoid sending spurious voltage changes to the computers when it is operated, it must be of the electronic type, and of course properly designed to meet the requirements outlined above.

So if the switch is disturbing the computer, the problem is the switch, and the best solution is to get a better one.

That being said, some keyboards, mice, motherboard I/O chips, and device driver software may be better than others at dealing with spurious signals from malfunctioning KVM switches. They may be designed to recognize and recover from such signals, perhaps delivering one or two bogus mouse movements, button pushes, or keystrokes, but not getting confused and locking up entirely, thus requiring a reboot or even a hardware reset. Mice and keyboards are cheaper than KVM switches, so one can try changing them (motherboards are not).

And if, e.g., the mouse driver has changed significantly between the 2.4 and 2.6 kernels, and the newer one doesn't perform as well with a bad KVM switch, one might examine the driver code and see if it has been revamped and some mechanism for recovering from spurious signals has been eliminated, and could be put back in. It's fairly easy to locate the pertinent code in the kernel sources, and presumably one could contact the relevant programmer, or post a message to the kernel mailing list, *if* one has carefully characterized the problem, and proved that it only happens with the new kernel version and not the old. But they may tell you to either get a good KVM switch, or do the reprogramming yourself. 8^)