OS/2 attack

Brad Silverberg

From: David Thacher
To: David Weise; Neil Konzen
Cc: attack; Richard Tong; Richard Tait
Subject: FW: RE: os/2 seamless windows
Date: Wednesday, March 11, 1992 2:07PM

Neil, thanks. One last request: can you point out for us a normal scenario using a very common app which used to cause problems in Win 3.1 which we now handle gracefully in Win 3.1 and which OS/2 still will fail on. Only by showing that in a demo can we prove this to the press and analysts.

David

>From neilk Wed Mar 11 13:11:24 1992
To: davidt davidw neilk
Cc: scottlu
Subject: RE: os/2 seamless windows

X-MSMail-MailClass: IPM.Microsoft Mail.Note
X-MSMail-Message-ID: CF2818D4
X-MSMail-Conversation-ID: CF2818D4
X-MSMail-Fixed-Font: 0001
X-MSMail-Priority: 0002
X-MSMail-WiseRemark: Microsoft Mail -- BETA 1
Date: Wed Mar 11 1992 13:09:25

Okay, here's what DavidW and I learned from screwing around with seamless windows:

Executive summary:

- They don't do any more "parameter validation" than we did in 3.0. 3.0 did do minimal validation, but it was quite easy to roach the system.

- If you crash or hang Windows, you only crash or hang that VM: other VMs (including other Windows VMs) continue to run.

- They perform some fairly intense synchronization at the GDI device driver level: semaphore locking using INT 33. This significantly increases the overhead of GDI calls, and is probably a major reason for their performance increase (but that's only a guess).

- Their system RIPs up a storm when running under the debug kernel. The error messages indicate potential semaphore timing windows in the kernel, "invalid window handle" rips, and lots of others. Sloppy code.

Plaintiff's Exhibit
7648
Comes V. Microsoft

Details:

For every Windows top-level window, there is a PM window that corresponds to it. Similarly, for every PM window there is a Windows window (though we did not check this out for sure: it was very difficult rooting around in Windows). This is how they accomplish clipping, z-ordering, focus, and activation between the two systems. With the help of a device driver, they perform RPC between Windows and PM whenever a top level Windows window is created, destroyed, moved, shown, etc: this RPC performs the analogous operation with the shadow PM window.

Their device drivers performed semaphore synchronization around most of the calls. We weren't able to understand this in too much detail, only that it existed.

When switching into the Windows VM, they execute a fair amount of synchronization and switching code in GetMessage/PeekMessage(). This probably primes the VM's system queue and such, but we weren't able to tell for sure.

For both GDI and USER, the only parameter validation code we could find was the code that already existed in 3.0, verbatim. Whenever they perform an operation on a top-level window, it is validated a little more strictly by OS/2 when they perform the operation on a mirror window, but this isn't a big deal. So, it's as easy to crash a Windows VM under OS/2 as it was to crash 3.0 -- it's no more reliable.

When the system boots and is running under the debug kernel, zillions of debugging messages are produced. Most of the messages indicate that the window manager semaphore is entered when it shouldn't be (implying potential for deadlock). This is tricky stuff, and easy for them to screw up. There were also quite a few "invalid window handle" messages and others. It seems like they don't even test with the debug systems, or, that the signal-to-noise ratio of the warnings is so low that they ignore the output.

- Neil