

tabbler

01/02/92

S83LTR

page 1

Date: Fri, 6 Oct 1989 14:55
Via: 1
To: LDBVAX::FKING
Cc: LDBVAX::FINGARI, COCONOR, DREED
Subject: VCPI Support In Windows 3

On September 28th Ben Williams and I met with members of the Microsoft Windows 3 development team to discuss the issues surrounding the support of VCPI in the 386 version of Windows 3. The essential technical problem is that Windows 3 for the 386 is currently designed as a control program that supports multiple virtual machines on top of a demand paged virtual memory system. All hardware and I/O for each machine is completely virtualized; that is, the control program mediates all access to memory and system devices, including the timer, by forcing the software in the virtual machines to run in v8086 mode in ring 3. However, a VCPI-based application such as 1-2-3 Release 3 must be able to run in ring 0. This implies that I/O to devices such as the screen, the printer, and the system clock cannot be virtualized.

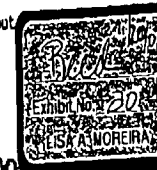
Clearly, supporting VCPI in each virtual machine would require a large change to the design of Windows 3. However, that is not what we asked for. We proposed that Windows be changed to support a single virtual DOS machine with access to VCPI. This machine would have the following restrictions:

1. No virtualization of the screen. This means that when active, the application runs in full screen mode, and when inactive and relegated to a window the application is frozen.
2. No virtualization of the system clock. This means that before starting the VCPI machine, Windows would have to reset the clock to run at the normal 18Hz rate.
3. No virtualization of the printer. This means that Windows would either have to prevent the VCPI machine from using the printer, or dedicate the printer only to the VCPI machine.
4. No virtualization of memory. All the memory belonging to the VCPI machine would have to be resident; no page faults would be allowed while it was running. This means that Windows would have to pre-page all of the machine's memory before starting it up.

In order to support the VCPI machine, Windows would have to add some code to set up the machine when the user switched to it, and to take it down when the user switched away, saving and restoring memory and screen images as required. In addition, it would have to monitor the VCPI calls that switch from protected to real mode in order to examine the interrupt controller and detect interrupts that needed to be reflected to other virtual machines, and hot keys that request a switch to another machine. All in all, this seems to Ben and me like a "bag" of a few hundred lines of code on the side of Windows that could later be removed, and not a radical restructuring of the product.

It's somewhat interesting to note that we faced a similar problem during the development of VM/370 in 1971. We began with a control program that provided complete virtualization for each user; however, in order to obtain reasonable performance for extended mode MVS machines we had to add the notion of a "privileged" virtual machine that had direct access to memory and certain dedicated I/O devices. So, the problem has been faced and solved before.

Perhaps more relevantly, we can always recommend that our users use DesqView/386; I understand that Bill goes non-linear when he hears about Quarterdeck.



001000

for Carl J. Young

LOTUS