Filling in some gaps in the story of Space Cadet Pinball on 64-bit Windows

devblogs.microsoft.com/oldnewthing/20220106-00

January 6, 2022

Raymond Chen

Space Cadet Pinball has a special place in the hearts of many Windows enthusiasts. A customer used their support contract to ask how to change among the three levels of play in Space Cadet Pinball. My proudest achievement of Windows XP was fixing the game so it didn't consume 100% CPU. People keep asking if it can be brought back.

One point of contention is over my claim that I removed Pinball from Windows because I couldn't get the 64-bit version to work. Retrocomputing enthusiast NCommander even undertook a Zapruder-level analysis of all of the 64-bit versions of Windows he could find to prove or disprove my story.

I was amazed at the level of thoroughness (and the fortitude it required to get those Itanium systems up and running, much less debug them), but there's one version of 64-bit Windows that NCommander didn't try out, and that's the one that's relevant to the story.

When the 64-bit Windows project started, there was no Itanium hardware yet. The only way you could run any Itanium code was to run it in a simulator. Booting Windows on an Itanium simulator took forever. Clearly not the development environment you wanted when porting millions of lines of code.

On the other hand, the Windows team did have access to a lot of copies of another 64-bit processor: The Alpha AXP.

In 1999, Compaq announced that it would no longer support Windows on the Alpha AXP, which left a lot of Windows team members with cumbersome paperweights on their desks.

Let's see what we've got here.

- A lot of Alpha AXP machines sitting around with nothing to do.
- A bunch of developers who have experience with the Alpha AXP.
- A 32-bit version of Windows that runs on the Alpha AXP.
- No Itanium hardware on the immediate horizon.
- A ticking clock.

The Alpha AXP is internally a 64-bit processor. It's just that 32-bit Windows used only the 32-bit subset.

Solution: Port the Alpha AXP version of 32-bit Windows to 64-bit Windows.

Now, 64-bit Windows on the Alpha AXP would never ship. But the Alpha AXP did have the advantage of *existing in physical form*, so the system could finish booting before the heat death of the universe. The assumption is that most of the effort in porting Windows to the Itanium is in the 32-bit to 64-bit transition, and not in dealing with quirks of the specific 64-bit processor you are porting to.

The assumption was somewhat validated by experience: The 32-bit Windows code base had been ported to many 32-bit processors, with relatively few architecture-specific issues. Once you got a 64-bit version of Windows working for the Alpha AXP, it should be a comparatively small amount of additional work to port it to Itanium. The hard part was going from 32-bit to 64-bit.

The team set to work, and we had 64-bit Windows running on physical Alpha AXP hardware long before we had any physical Itanium hardware. I could test my 64-bit port on a physical Alpha AXP system to validate that it was successful. And that's the system that had the broken collision detector.

NCommander did find a collision detection bug on the Itanium, although that bug was nowhere as severe as the one that existed on the Alpha AXP. My guess is that it had to do with the default rounding mode established by the C runtime library.

My theory as to what happened is that some time after I removed Pinball from the product, the C runtime team realized that they had a compatibility bug in the way they set the default rounding mode, and they fixed it. Or maybe there was a compiler bug, and the compiler team fixed that. Whatever the problem was, somebody fixed it, and then they went back and re-tested Pinball with this fix, and everything worked great, so they put Pinball back.

I'm just guessing about what happened afterward because nobody informed me that they had gotten Pinball working and added it back. I just assumed that it was gone forever.

Raymond Chen

Follow