The Vx-Files:

What the Media Couldn't Tell You About Mars Pathfinder

by **Tom Durkin**

Arguably the most spectacular interplanetary robot mission in history, Mars Pathfinder outperformed all expectations – and ironically, that was why the lander developed a mysterious communications problem shortly after its successful landing July 4, 1997.

photo courtesy of NASA

For no apparent reason, *Pathfinder's* onboard computer would spontaneously reset itself. This happened about a half dozen times in the first few weeks after the landing.

Contrary to press reports at the time, no data was lost, but data collection was delayed by a day every time *Pathfinder* reset itself.

"Reports of the problem in the popular press were generally incoherent, and in some cases wildly wrong," according to David Wilner, chief technical officer of Wind River Systems of Alameda, Calif.

In a Dec. 3, 1997, keynote address to the 18th IEEE Real-Time Systems Symposium in San Francisco, Wilner explained that the media couldn't grasp the complexity of the problem – much less the answer – "because of the difficulty of explaining the issues to non-engineers."

However, because he was speaking to real engineers Dec. 3, Wilner detailed precisely what went wrong last summer, why it went wrong – and how it got fixed. Wilner was uniquely qualified to speak on the subject, because Wind River makes the RTOS (real-time operating system) – VxWorks – which was embedded in *Pathfinder's* onboard computer.

Fortunately, Wind River and NASA software engineers were able to replicate the problem on a duplicate of *Pathfinder* at the Jet Propulsion Laboratory in Pasadena, Calif. It only took 18 hours to fix the problem – but it took almost three weeks to find it.

People Will Talk

Michael Jones, a researcher in the Operating Systems Research Group at Microsoft Corp., found Wilner's speech so interesting that he wrote up a report on it and e-mailed it to some friends, colleagues and educators.

Within days, his report was posted on several Usenet interest groups, where it was widely redistributed throughout the larger Internet community.

"It was actually a pretty good take on my description," Wilner told *RS&T*, adding, "One of the guys at JPL posted a reply to Jones that went into a lot more gory detail about what happened from their point of view."

The "guy" at JPL was Glenn Reeves – the flight software cognizant engineer for *Mars Pathfinder*. Reeves led the team that identified *Pathfinder's* reset problem and fixed it. Using Jones' report as a framework for his response, Reeves elaborated – in painstaking detail – the exact software engineering methodology his team went through to replicate, and then rectify, what the general media called a "software glitch."

Flight software engineer Glenn Reeves had reason to grin after he and his team solved Pathfinder's *"software glitch." Behind him is the duplicate Mars lander that the JPL/Wind River team used to find and fix the priority inversion problem in the VxWorks® program. "The biggest thanks should go to the software team that I had the privilege of leading. Their expertise allowed us to succeed," he emphasized.*

How VxWorks Works

In order to understand what went wrong, it is necessary to understand what VxWorks does. Basically, the software provides "preemptive priority scheduling" of data threads and mission control commands.

Data (video images, soil samples, meteorological readings, etc.) from the various instruments on the lander (*Pathfinder*) and the rover (*Sojourner*) had to pass through an information bus in *Pathfinder's* computer to be transmitted to Earth.

Likewise, commands to control the devices on *Pathfinder* and *Sojourner* (such as the cameras or alpha proton X-ray spectrometer) had to move through the same information bus in the opposite direction.

Obviously, this couldn't happen all at once. Data threads and command strings had to take turns using the bus. Furthermore, some data and commands were more important than others.

Thus, it was the job of VxWorks to schedule traffic through the bus according to the pre-assigned priorities of data and commands.

In order to prevent communications conflicts, VxWorks synchronized access to the bus with mutual exclusion software locks known as "mutexes" or "semaphores." When a specific task was running, its semaphore blocked other tasks from interfering with it.

"A Classic Case of Priority Inversion"

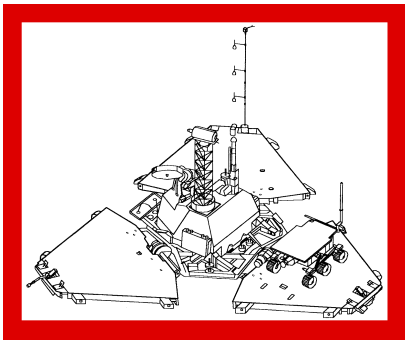
"Most of the time this combination worked fine," Jones wrote in his report. However, very infrequently, an interrupt was sent to the bus that caused a medium-priority communications task to be scheduled during the split-second interval when a high-priority thread was blocked while waiting for a low-priority meteorological data thread to run.

In this case, Jones explained, the long-running, medium-priority communications task – having a higher priority than the low-priority meteorological task – would prevent the meteorological task from running. After a predetermined time had passed, a watchdog timer would go off, notice that the low-priority data bus task had not been executed on time, conclude that something had gone wrong, and initiate a total system reset.

"This scenario is a classic case of priority inversion," Jones asserted. "Priority inversion is a difficult concept to explain," Reeves told *RS&T*. "It doesn't make any sense in everyday terms. In a nutshell, what happened is there was a situation where the act of taking the information took longer than the amount of time we had allotted for the rest of the communications to occur within the software time limits."

Too Much of a Good Thing

What caused the priority inversion was that *Pathfinder's* antenna performed better than expected.



"It turned out that we got a much higher meteorological data rate, because we could point the antenna at Earth much better than we ever imagined," Reeves said. "We didn't expect it. We had never actually tested the thing with that high a set of data rates. It was better than the best possible hope we had.

"It was really great news from the spacecraft, but it was somewhat disconcerting to us, because it did expose a bug that we really wish we had caught."

Bug Busters

"The software that flies on *Mars Pathfinder* has several debug features within it that are used in the lab but are not used on the flight spacecraft (not used because some of them produce more information than we can send back to Earth),**"** Reeves wrote in his response to Jones.

While not enabled on the spacecraft, these features remained in the software by design. "We strongly believe in the test-what-you-fly-and-fly-what-you-test philosophy."

One of the debugging tools was a trace/log facility that was originally developed to find a bug in an early version of VxWorks.

"After the problem occurred on Mars, we ran the same set of activities over and over again in the lab," Reeves said. The trace/log was coded to dump its record of activities when a failure occurred. Finally, "we were able to cause the problem to occur. Once we were able to reproduce the failure, the priority inversion problem was obvious."

What was not so obvious was the solution ...

The Big Fix

As both Jones and Reeves explained it, the "priority inheritance" parameter of the semaphore for the meteorological data thread was not enabled in *Pathfinder's* VxWorks software.

If priority inheritance had been enabled, "the low-priority meteorological thread would have inherited the priority of the high-priority data bus thread blocked on it while it held the mutex, causing it be scheduled with higher priority than the medium-priority communications task, thus preventing the priority inversion." Jones wrote.

Clearly, the JPL flight software team had to "change the creation flags for the semaphore so as to enable the priority inheritance," Reeves explained.

There were, however, several potential problems with enabling priority inheritance on the meteorological data thread semaphore.

First, there was no way to enable the inheritance parameter only on the semaphore for the meteorological thread. Enabling the meteorological semaphore would enable all of the other semaphores in the program as well. How would this change the behavior of the rest of the system?

Second, Wind River had deliberately turned off the priority inheritance option before launch to optimize the performance of VxWorks. Would performance be degraded if it were turned on?

After intense consultations with Wind River personnel and extensive testing on the duplicate *Pathfinder* in the JPL lab, the flight software team determined that there would be no adverse impact by activating the priority inheritance function.

Repair by Remote Control

"Patching " software on a spacecraft on another planet is a somewhat specialized process.

As Reeves explained it, the team had to transmit the differences between what was onboard *Pathfinder* and what was onboard the reconfigured replica in the lab. They used custom software ("with a whole bunch of validation") that was already onboard *Pathfinder* to modify the VxWorks operating system.

The ending was somewhat anticlimactic. Buried in a July 21 JPL press release was this aside: "[The flight software team] also sent a software update to correct sequences onboard the flight computer which have caused it to automatically reset itself."

When asked for any final comments on the priority inversion problem, Reeves said, "Even when you think you've tested everything that you can possibly imagine, you're *wrong*."

Although he broke the news on the program malfunction, Jones stressed, "I greatly admire what the *Pathfinder* team accomplished."

At JPL, Reeves echoed the same sentiment: "The team, dedicated to a single purpose, is what got this thing done. They're the ones who really deserve the credit for making it happen. The Wind River guys did a really good job."

RS&T