Elisha Parslow

CS315 Distributed Scalable Computing

HW3 Part 3

Mars Rover

The article about the Mars Pathfinder mission addresses what was the cause of the total system reset that occurred a few days into the mission when the rover was landed. The source of this reset was due to threads reaching a mutex lock at different times then their priority. A medium priority thread would get to the critical piece of code and hold onto it while the high priority had to wait. The system would use a watchdog timer that eventually opens the mutex lock after a set amount of time and it would realize the high priority thread was not executed, concluding something had gone wrong, and completely reset the system. This is known as a priority inversion.

The VxWorks team was able to debug this system reset through running the recorded actions of the code on a replica of the rover in a lab. They eventually were able to recreate the bug and discovered it was a priority inversion. The engineers corrected the bug by using priority inversion where the program recognized priority through a parameter that they activated. They uploaded the update to the spacecraft and no more resets occurred.

The learnings from this incident were to leave debugging in the system, otherwise the problem would have never been solved and to use priority inheritance when a data bus is set to be time critical. In the pre-launch, there had actually been two incidents when a system reset occurred and the engineers were never able to locate it and thought it would not happen when launched. This is important for realizing the importance of priority bugs.