

As falhas numéricas que podem causar desastres

4 de junho de 1996 será para sempre lembrado como um dia sombrio para a Agência Espacial Europeia. O primeiro voo não tripulado do foguete Ariane 5, que decolou carregando quatro satélites científicos caríssimos, acabou 39 segundos depois do lançamento, em uma horrível bola de fogo e fumaça. Estima-se que a explosão causou um prejuízo de US\$ 370 milhões.

Não foi uma falha mecânica nem um ato de sabotagem. O desastre foi causado por um simples *bug* em um software, que fez cálculos errados ao se tornar sobrecarregado com números mais longos do que era capaz de suportar.

Erros semelhantes foram também os responsáveis por uma série de incidentes nos últimos anos, fazendo sondas espaciais desaparecerem ou desviando mísseis de seus alvos.

Mas como é possível que computadores fiquem tão embaralhados com números? E o que causa esses *bugs*?

[Leia mais: Quando encontraremos seres alienígenas?](#)

[Leia mais: Será que um dia viveremos debaixo da terra?](#)

'Virada' perigosa

Imagine tentar representar um valor de, por exemplo, 105.350 quilômetros em um hodômetro que tem um valor máximo de 99.999. O contador "viraria" para 00.000 e contaria até 5.350, a quantia restante. Foi esse mesmo tipo de imprecisão que condenou o lançamento do Ariane 5.

Tecnicamente, o problema é chamado de "estouro de inteiros", o que significa que os números são muito longos para serem armazenados em um sistema computacional, às vezes provocando seu mau funcionamento.

Uma investigação do incidente com o foguete europeu revelou que um processo deixado pelo software da geração anterior, o Ariane 4, capturou uma inesperada medida de velocidade no novo veículo – muito mais rápido que seu antecessor – e o software do Ariane 5 não conseguiu lidar com esse número tão grande.

Uma sequência de autodestruição foi iniciada e, segundos depois, acontecia a tragédia.

Esse tipo de falha ocorre com uma frequência surpreendente. Acredita-se que um dos motivos pelo qual a Nasa perdeu o contato com a sonda espacial Deep Impact, em 2013, foi o fato de seu software ter alcançado um inteiro.

A ameaça de 2.147.483.647

Direito de imagem Getty Image caption Especialistas temem bug que poderá causar panes em janeiro de 2038

No início de maio, surgiu a notícia de que um Boeing 787 poderá sofrer um problema semelhante: a unidade de controle que gerencia a distribuição de energia para os motores do avião pode entrar automaticamente em um modo à prova de falhas – desligando as turbinas – se for deixada ligada por mais de 284 dias.

Hipoteticamente, os motores poderiam parar repentinamente em pleno voo.

A diretriz da Administração Federal de Aviação dos Estados Unidos em relação à questão estabelece que um contador no software da unidade de controle "estoure" depois desse período específico de tempo.

Apesar de a FAA e a Boeing terem se recusado a comentar o assunto para esta reportagem, alguns observadores amadores destacaram que, quando contados em centésimos de segundo, 248 dias equivalem ao número 2.147.483.647 – o que é um dado muito importante.

Isso porque 2.147.483.647 é o valor positivo máximo que pode ser armazenado por um registrador de 32 bits, instalado em muitos sistemas computacionais.

No caso do Ariane 5, o software estava usando um espaço de 16 bits, capaz de armazenar o valor máximo de 32.767.

[Leia mais: Moda viral: as redes sociais acabarão com as passarelas?](#)

'Economia' de dados

Mas se os números são infinitos, por que usamos espaços de armazenagem tão limitados? Basta lembrarmos que os computadores tradicionalmente demandam eficiência em todas as frentes.

Quando os primeiros modelos surgiram, o espaço de armazenamento costumava era bem mais caro e o processamento de grandes números era mais lento. Mas, mantendo-se alguns limites, como os contadores, os softwares rodavam com mais facilidade.

Sistemas de foguetes teleguiados processam uma quantidade enorme de números a uma grande velocidade, por isso essa capacidade realmente importa. Mas até o desastre com o Ariane 5, essas limitações não eram vistas como problemáticas.

"Temos que reconhecer que, em termos de software, estamos sempre aproximando a realidade", explica Bill Scherlis, especialista da Universidade Carnegie Mellon, nos Estados Unidos. "A

engenharia precisa encontrar um ponto de equilíbrio entre o custo de se ter uma contagem mais precisa e a eficiência com que o programa roda."