

案例：浮点运算不精确产生的灾难

（软件缺陷的可能会发生在任何地方，甚至是一些简单的运算中）

1991年2月25日，在海湾战争期间，沙特阿拉伯的摩达地区设置的美国爱国者导弹，拦截伊拉克的飞毛腿导弹失败。飞毛腿导弹击中了美国的一个兵营，造成28名士兵死亡。美国总审计局（GAO）对失败原因做了详细的分析，并且确定潜在的原因在于一个数据计算不精确。

爱国者导弹系统中含有一个内置的时钟，实现为一个计数器，每0.1秒就加1。为了以秒为单位来确定时间，程序将用一个24位的近似1/10的二进制小数值来乘以这个计数器的值。特别地，1/10的二进制表达式是个无穷序列

0.000110011[0011]...

其中，方括号里的部分是无限重复的。计算机只用这个序列的开头位和二进制小数点右边的头23位来近似的表示0.1。我们称这个数为x，那么：

- A. x-0.1的二进制表示是什么？
- B. x-0.1的近似十进制值是什么？
- C. 当系统启动时，时钟从0开始，并且一直保持计数。在这个例子里，系统已经运行了大约100个小时。程序计算的时间和实际的时间之差为多少？
- D. 系统根据一枚来袭的导弹的速率和它最后被雷达侦测到的时间，来预测它将在哪里出现。假定飞毛腿的速率大约是2000米/秒，对它的预测偏差了多少？

正常地，一个通过一次读取时钟得到的绝对时间中的轻微错误不会影响跟踪的计算。反而，它应该依赖于两次连续的读取之间的相对时间。问题是爱国者导弹的软件已经升级成使用更精确的函数来读取时间，但是不是所有的函数调用都用新的代码替换了。结果就是，跟踪软件使用了一次读取的精确时间，但其他软件读取的不是精确时间。