THE CLOCK GROWS AT MIDNIGHT

INSIDE RISKS

Peter G. Neumann

Background

Roosters are sometimes more predictable than computer clocks, for a variety of reasons. For the new year, we focus on clock-related problems.

Overflows

The number N = 32,768 = 2^{15} has caused all sorts of grief resulting from the overflow of a 16-bit word. A Washington DC hospital computer collapsed on 1989 Sep 19, N days after 1900 Jan 01, forcing a lengthy period of manual operation. Brian Randell reported that the University of Newcastle upon Tyne, England had a Michigan Terminal System (MTS) that crashed on 1989 Nov 16, N days after 1900 Mar 01. Five hours later, MTS installations on the East Coast of the U.S. died, followed by others across the country—an example of a genuine (but unintentional) distributed time bomb.

John McLeod noted that COBOL uses a two-character date field, and warned about having money in the bank at midnight on 1999 Dec 31. Robert I. Eachus noted that the Ada *time of year* field blows up after 2099, and MS-DOS bellies up on 2048 Jan 01.

Year-end roundup

The Pennsylvania Wild Card Lotto computer system lassoed itself on 1990 Jan 01, and winners of the lottery three days later could not be determined until the software had been patched.

Leaping forward

John Knight reported that a Shuttle launch scheduled to cross the end-of-year 1989 was delayed in order to avoid the software risks of a) the year-end rollover and b) a leap-second correction.

Not making ends meat

Shortly after 1988 Feb 29, the Xtra supermarket was fined $1,000 for keeping meat one day too long, because the computer program was not making the adjustment for leap year.

Arithmetic errors

John Knight found an item in the October 1990 *Continental* magazine (while flying home from the Las Cruces safety workshop), describing how the airline rents aircraft by the day and consistently charged one day too few because they merely subtracted the dates.

Dependence on remote clocks

In Colorado Springs, a child was killed and another was injured at a traffic crossing, when the school-schedule-dependent computer controlling the street crossing did not properly receive the time transmitted by the atomic clock in Boulder.

SRI's Computer Science Laboratory computer system once used a then-common eleven-clock averaging algorithm to reset the local clock automatically after a crash. Unfortunately, at the moment of reboot, a clock at the University of Maryland was off by twelve years, based upon which the CSL clock was initialized to be off by 15 months. (Yes, the *new* algorithms discard extreme values, and rely on systems with more dependable clocks.)

Byzantine clocks

Algorithms for making a reliable clock out of less reliable clocks are potentially nontrivial. The old three-clock algorithms (e.g., take the middle value) break down if one clock can report different values to its neighbors at any one time. In that case, four clocks are required, or $3n + 1$ if n are potentially untrustworthy [1, 2].

Reactions in RISKS

Contributions on this topic have spanned numerous types of clock problems. Most everyone seems concerned about the coming millenium, and speculations on what might happen are rampant. But there are many other problems lurking, some suggested here.

Conclusions

The maximum-field-value problem would seem obvious enough to have been better anticipated; however, it keeps recurring. Distributed system clocks are a potential source of serious difficulties, as illustrated by the synchronization problem that caused postponement of the first Shuttle launch [3]. Defensive design is particularly appropriate.

"Planning for the future" is usually important. It is high time that we looked further ahead. We have nine years to think about the millenial problems, analyzing existing programs and developing standard algorithms that can help avoid alarming consequences. ■

References

1. Lamport, L., and Melliar-Smith, P.M. Synchronizing clocks in the presence of faults. *JACM 32,* 1 (Jan. 1985), 52–78.
2. Rushby, J.M., and von Henke, F. Formal Verification of the Interactive Convergence Clock Synchronization Algorithm using EHDM. SRI-CSL-89-3. SRI International, Menlo Park CA, February 1989.
3. Garman, J.R. The bug heard 'round the world. *ACM Softw. Eng. Not. 6,* 5 (October 1981), 3–10.
4. Other cases noted above were discussed in "RISKS" and in *SEN* 11, 2; *13,* 2; *14,* 2; *14,* 6; and *15,* 1.

Peter G. Neumann is chairman of the ACM Committee on Computers and Public Policy, moderator of the ACM Forum on Risks to the Public in the Use of Computers and Related Systems, and editor of ACM SIGSOFT's *SEN.* Contact risks-request@csl.sci.com for on-line receipt of RISKS.