NRAO Y2K

The PC Clock/Date Problem

Alan Bridle
NRAO Charlottesville

Summary

Most PCs sold before mid-1997 contain clocks that will not update the century at their first reboot after 31 December 1999 without (one-time) human intervention. Furthermore, some PC BIOS may still report incorrect years to applications that interrogate their clocks, even after the century has been set correctly. For many PC users, a simple one-time setting of the century information using the os `date` command will work around the main problem. But any NRAO PC+software combinations that do date-aware accounting, control or forecasting, or which pass their internal date stamps to other machines or into mission-critical databases (at the NRAO or its contractors) should be examined closely for Y2K compliance.

Introduction

I began taking "Y2K" seriously when I realised that it was not just a problem with dusty-deck COBOL code in mainframe machines ... a nasty problem for banks, insurance companies, financial houses and most government agencies ... but it was sitting in front of me in my PC, whose real time clock would will not update correctly to 2000 and whose BIOS may behave strangely when the year hits '00'.

The problem starts with the battery-backed Motorola MC146818 Real Time Clock plus RAM (RTC) chip that is used (or emulated) for maintaining date and time when a PC is turned off. The earliest (pre-AT) IBM PC's and clones had no RTC. You set the date and time manually every time they were booted.

The Motorola MC146818 uses only 8 bits to store the date, sufficient to hold only the last two digits of the year in packed BCD format. When the year goes past 99, it rolls over to 00.

As the MC 146818 contains non-volatile memory, IBM specified that location `32h` would hold the century information. By combining the year from the clock with a "century bit" from `32h`, a four-digit year could be passed to the operating system at boot-up and to applications. But this externalised century information is not manipulated by the clock. When the year rolls from 99 to 00, nothing changes at `32h`!

It got worse when PS/2 was introduced. IBM specified location `37h` as the placeholder for the century, not `32h`.

PC software can obtain the date from the RTC via BIOS interrupt `1Ah`, function `04h`, or by reading the RTC directly through I/O ports `70h` and `71h`. Any program that bypasses the BIOS to access the RTC directly must know whether the PC conforms to the `32h` standard or the `37h` standard. Any program that tries to be compatible with both the "AT" and the "PS/2" standards will therefore use the BIOS interrupt to obtain the date.

That's where all hell can break loose. The date-sensitive behavior of the entire PC clock+BIOS+application system depends on the state of the hardware clock, how the particular BIOS was coded, and how the application itself was designed. Whether a given situation produces a "Y2K problem" for real-time date stamps depends on all three ingredients and how they co-operate.

Many BIOS respond to '00' in the year from the RTC and '19' in the century bit by treating the inferred '1900' date as an error. PC's did not exist in 1900! So the BIOS is designed to subsitute a "legacy date" appropriate to the earliest Intel chips ... 4 January 1980 in many systems, 1 January 1980 in some, 1 January 1900 in others. This condition persists until the *user* sets the century bit to '20', via the operating system's `date` command. A few BIOS do not permit this setting, but most will allow it and will then perform sensibly thereafter.

So what happens on 1 January 2000?

If a non-compliant PC is left powered on as the century rolls over (and the power doesn't go off at midnight from a Y2K failure in the generating system!) the PC may appear to soldier on happily until its next reboot. The operating system maintains time and date independently of the RTC once the machine has been booted, by counting cycles as an offset from a fixed time and date. So the operating system's "clock" may roll over into the new century until the next time the machine is re-booted. At that time, the RTC year setting is consulted again and the date (typically) goes haywire until it reset by the user.

The first reboot of a non-compliant PC in January 2000 will have to be followed by a `date` command that sets the year to 2000, just like every reboot on an old pre-AT PC. If the PC begins launching tasks automatically at reboot, the normal reboot sequence may need to be replaced or interrupted in order to reset the century information in the RTC.

Once the century bit in the RTC has been set to '20', it will be retained. Future reboots will correctly infer the 20+yy year and no further action is needed to maintain the RTC settings.

A few BIOS won't allow the year to be set beyond 1999. These are mainly in very old machines that should be obsoleted anyway.

There are TSR programs available to work around the BIOS's response to the '00' year, but it will likely take more time to get and instal such TSR's than to execute the `date` command once! The merit of the TSR approach is that the patch can be installed before January 2000 and a normal boot sequence followed thereafter. The user does not have to remember to do anything unusual at the first reboot. A TSR could also correct the problem again if the century bit gets reset back to '19' for any reason. A minor danger is that the TSR could be bypassed or deleted accidentally.

Another fix is for the PC to obtain the current date and time and reset its clock from a master source over a network. This is fine so long as the network is available, the remote time source itself Y2K-compliant, and the network's clock-correcting software can handle large apparent offsets

(potentially 20-100 years!)

Problems after the clock is set correctly

The "catch" is that even after the user, a network, or a TSR have set the century bit to '20' after a reboot, a badly-behaved non-Y2K compliant BIOS can still feed an erroneous date to an application, or behave strangely in other ways, when there is a '00' in the RTC's year field.

Software packages that use the BIOS interrupt to get a real-time date stamp may still acquire an incorrect date if the BIOS does not account for the actual setting of the century bit when it handles the `1Ah` interrupt.

The worst example I have seen reported on the net was a BIOS which allowed the century bit to be set to '20' and the machine to run semi-normally for several days, but then eventually overflowed a memory register and began corrupting system configuration information. This computer's second serial port became inoperative after about 10 days in a Y2K test, but reappeared when the date was set back to 1997.

Applications

Even when the RTC and the BIOS are doing sensible things, the software that runs in the PC may be non-compliant internally. I have done a number of Y2K tests by advancing the system date on an old Zeos 386 that runs Windows 3.1 and MSDOS 6.2. Even after the RTC and CMOS setting were apparently all set for the next millennium, two of three File Managers that I tested turned out not to be Y2K compliant. The Windows 3.1 File Manager reports the year of a newly-created file as 19:0 for example, not 2000. (Microsoft says the File Manager is not part of the operating system, and asserts that Windows 3.1 is compliant). A powerful directory utility that I've used for years under DOS reported the date on a 1-1-2000 file as 01-01-10.

In these cases, the clocks and the date stamps in the FAT for newly-written files were all correct and other utilities and programs handled them correctly, but these utilities interrogate them and/or display them in non-Y2K compliant ways.

For other applications such as spreadsheets and databases the Y2K issues include whether a YYYY format is used for the year internally, or how two-digit years are windowed into centuries if not. Also whether an application correctly recognizes that 2000 is a Leap Year, so that 29-February-2000 is a valid date.

Some spreadsheet packages, e.g. Microsoft *Excel for Windows 95 V7.0*, Lotus *1-2-3 V5.01*, accept 29-February-2000 as valid but also (incorrectly) accept 29-February-1900 as valid. This is of course a symptom that their embedded Leap Year algorithms are *still* incorrect, but seems unlikely to impact key processes at the NRAO! Quattro pro vsn 7 seems to handle leap years correctly.

We can probably continue to use many older, non-compliant PC's after 1 January 2000 with a modest amount of user education and alertness, plus a (generally one-time) reset of the century bit. Stand-alone systems that do no date-aware calculations or control, e.g. PC s used for word-processing or minor number crunching in a scientist's office, or in an off-line engineering testbed, ~~we~~ can probably keep ~~them~~ going in the hands of careful users until we would obsolete them for other reasons.

But PC's that write real-time date stamps into files or databases that are exchanged with other computers, at the NRAO or elsewhere, especially in mission-critical applications like control of machinery, or for business and personnel operations, should probably be upgraded to systems with certified and tested Y2K-compliant clocks, Y2K-compliant BIOS and Y2K-compliant applications software.

The number of different ways in which the real-time clock, BIOS and an application can interact to generate incorrect date stamps in non-compliant PC systems is large. The investment of time needed to test all possible operational situations on all older systems may drain more from the NRAO budget than a simple replacement of a mission-critical system with a newer, compliant one.

Recommendation

I suggest that we should first identify where any PC's are used for date-aware functions that are critical to the overall mission of the NRAO, and make simple tests of their Y2K compliance using available test packages.

Testing of Y2K compliance in systems with licensed software packages or time-aware file deletion or backup utilities may create significant problems. Users should back up their entire disk structures before attempting roll-forward tests, in case it is not possible simply to roll back to the original state at the end of the test!

Any NRAO PC's that are truly mission-critical (the meaning of this this must be defined on a site-by-site, division-by-division basis) and which are found to be non-compliant should be upgraded to compliance (e.g. if the BIOS can be flashed), or replaced.

Any that are not mission-critical may not need to be replaced, but could be used with care and moderate user awareness of the potential for subtle Y2K-related failures, until they are "retired" for other reasons.

For PC's that do not do any date-aware calculations, the main issue may simply be file management, backup and book-keeping in the presence of dates that may be displayed or sorted improperly by some utilities. With a little user education we can probably let many staff determine for themselves whether there any show-stoppers from Y2K bugs in their NRAO PCs, while we concentrate resources on the observatory-wide mission-critical areas such as telescope operations, business and personnel services.

Last updated: 5 September1997 at 15:10 EDT
abridle@nrao.edu

add somewhere that in general it is not a good idea to leave a PC running through the century change, unless there is a strong reason to do so.