

Mille e non più mille Linux e l'anno 2000

Mancano sempre meno giorni al 2000 e negli informatici cresce sempre più la curiosità di vedere quali saranno i reali effetti di quello che è stato battezzato il "bug del millennio".

Nonostante il problema sia conosciuto da anni, pochi infatti si sono preoccupati di risolverlo per tempo. In questo articolo vedremo alcuni suggerimenti sulle precauzioni da prendere con Linux e come verificare che il nostro sistema possa continuare a lavorare in piena efficienza... almeno per altri 38 anni.

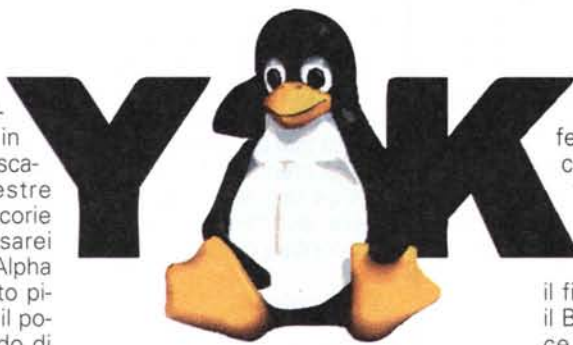
di Giuseppe Zanetti

Più che all'anno 2000, le mie aspettative di bambino erano rivolte al 13 settembre del 1999, data in cui la luna avrebbe dovuto essere scagliata fuori dall'orbita terrestre dall'esplosione di un deposito di scorie nucleari. Ero infatti convinto che sarei stato uno degli abitanti della base Alpha e che magari un giorno avrei potuto pilotare un'Aquila o magari prendere il posto del capitano Koenig al comando di quel manipolo di trecento abitanti.

Per i pochissimi lettori che ancora non lo sapessero, ricordo che con il termine "bug dell'anno 2000" (Y2K, dove la Y è l'abbreviazione di Year e K indica le migliaia, come nei Kg) si intende far riferimento a tutta una serie di problematiche legate alla difficoltà di alcuni vecchi (e purtroppo anche recenti) programmi nel riconoscere correttamente il passaggio dal 31 dicembre 1999 al primo gennaio del 2000 (non oso dire "il cambio di millennio" per non attirarmi gli strali di quelli che sono convinti che il millennio termini alla fine del 2000).

La causa: la poca lungimiranza

La causa del problema deve ricercarsi indietro negli anni, e precisamente



quando, per risparmiare pochi byte di memoria, a quel tempo risorsa molto preziosa, si decise di codificare le date indicando solamente le ultime cifre dell'anno (99 al posto di 1999). Il problema viene alla luce quando si tentano di sottrarre date posteriori al 2000 con date ad esso antecedenti: in questo caso infatti la sottrazione 00 - 99 restituisce un numero negativo. A seconda dei casi questo può comportare una semplice scocciatura, ma anche il possibile annullamento del nostro conto in banca o la scelta di buttare via una partita nuova di medicinali o di generi alimentari come se fossero ormai scaduti da un secolo (oppure, che forse è ancora peggio, di mettere in circolazione merce ormai vecchia).

Si noti che si tratta di un problema di progetto dovuto alla poca lungimiranza di alcuni programmatori. Il fatto che si tratti di un errore di progetto non signifi-

ca perciò che tutte le implementazioni ne siano necessariamente affette. È necessario verificare caso per caso, programma per programma che tutto funzioni alla perfezione.

Il problema non è limitato ai soli sistemi operativi e programmi applicativi, ma può riguardare anche il firmware delle macchine (ad esempio il BIOS dei personal computer) o il codice contenuto nelle ROM dei sistemi "embedded". E questo è forse l'aspetto più grave, in quanto non solo siamo portati a dimenticare che anche il videoregistratore, l'orologio da tavolo, la lavatrice o l'ascensore contengono al loro interno dei microprocessori e dei programmi, ma anche perché spesso l'aggiornamento del software difettoso in questo tipo di sistemi è una operazione molto meno agevole rispetto al computer di casa.

I diversi settori dell'industria e dell'economia, primo fra tutti quello bancario, hanno già provveduto o stanno provvedendo ad affrontare il problema, con più o meno urgenza a seconda dell'importanza delle eventuali conseguenze. Così, a meno di eventuali sorprese, non vale la pena preoccuparsi più di tanto per il proprio conto corrente o per la carta di credito. Se le banche sono passate indenni attraverso l'Euro, figuriamoci che paura possono avere di un problema che si co-

nosce da più di 30 anni.

È bene invece porre una certa attenzione a tutti quei sistemi che sono governati da molte variabili e che possono andare facilmente in crisi, come i trasporti. Quasi sicuramente a causa del bug del 2000 non cadrà nessun aereo, ma probabilmente vi saranno dei ritardi o degli errori nelle prenotazioni dovuti a qualche computer andato in crisi. Perciò, se state pensando di prenotare un viaggio in Australia per vedere per primi l'alba del nuovo millennio, vi conviene prendere in considerazione l'ipotesi di allungare un po' la vacanza, piuttosto che passare quei giorni ad aspettare fermi in qualche aeroporto.

Per quanto riguarda gli apparecchi domestici, sarebbe bene fare un piccolo inventario ed andare a verificare nei siti dei diversi fabbricanti la compatibilità dei loro prodotti con l'anno Duemila. Molti videoregistratori ad esempio non sanno riconoscere che il 2000 è un anno bisestile e passano come se nulla fosse dal 28 febbraio al primo marzo. Un buon sito, dedicato in particolare al gentil sesso, che spiega le consequen-

ze che può avere il problema sulla vita di tutti i giorni e sulle famiglie è Y2K for Women (<http://www.y2kwomen.com/>).

Anno bisesto, anno senza sesto

Nonostante sia il più grave, quello del primo gennaio non è l'unico problema con cui dovremo fare i conti nel nuovo millennio. Non occorrerà neppure aspettare troppo, dato che già il 28 febbraio bisognerà stare attenti a verificare che i nostri computer si siano ricordati di trovarsi in un anno bisestile. La regola infatti dice che un anno è bisestile se è divisibile per 4 ma non è divisibile per 100. Alcuni programmatori dimenticano però una piccola postilla: l'anno è ancora bisestile se è divisibile per 400. Perciò il 2000 è bisestile.

Lo afferma lo stesso Papa Gregorio, riformatore del calendario, nell'*Inter Gravissimas*:

"Anno vero MM, more consueto dies

bissextus intercaletur, februario dies XXIX continente, idemque ordo intermittendi intercalandique bissextum diem in quadringentis quibusque annis perpetuo conservetur".

A proposito, sapreste rispondere con assoluta sicurezza alla domanda "dov'era il tristristrisavolo di Pippo (l'amico di Topolino, non il nome della variabile!) il giorno 10 settembre del 1752?". Semplice: non era da nessuna parte, in quanto tale data non è mai esistita. Il buon Papa Gregorio, infatti, per far quadrare il suo nuovo calendario con il moto della terra ha dovuto recuperare un po' di anni bisestili, a spese dei giorni dal 3 al 13. Potete sincerarvene scrivendo:

cal 9 1752

```

September 1752
S M Tu W Th F S
                1 2 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
    
```

Bookmark

Un ottimo punto di partenza per la navigazione alla ricerca di informazioni su Linux e il problema dell'anno 2000 è sicuramente <http://www.linux.org/help/beginner/year2000.html>.

Altri link interessanti sono:

http://www.linuxworld.com/linuxworld/lw-1998-10/lw-10-y2k.html	Linux in the New Millennium
http://www.nltug.org/~cbbrowne/linuxy2k.html	Linux and Year 2000 (Christopher Browne)
http://www.linux.org.uk/mbug.html	Linux and Year 2000 (UK.Linux.Org)
http://www.gnu.org/software/year2000.html	GNU Software in the Year 2000
http://www.gnu.org/software/year2000-list.html	List of GNU Software and Year 2000 Problems
http://language.perl.com/news/y2k.html	Year 2000 Compliance: Lawyers, Liars, and Perl
http://www.opengroup.org/desktop/faq/y2k.htm	Open Group Desktop Technologies in the Year 2000
http://www.apache.org/docs/misc/FAQ.html#year2000	Apache Year 2000 Statement
http://www.apacheweek.com/issues/97-05-23#y2k	Apache and the Year 2000
http://charlotte.redhat.com/legal_statement.phtml/#y2k	Redhat Year 2000 Readiness Disclosure
http://www.debian.org/News/1998/19980104/	Debian and the Millennium Bug
http://master.debian.org/~csmall/y2k.html	Debian and Y2K - technical information
http://www.suse.com/PressReleases/y2k.html	S.u.S.E. Linux and the Year 2000
http://www.suse.com/PressReleases/y2k-2.html	S.u.S.E. Linux and the Year 2000 - Update
http://www.applix.com/appware/aa_y2k.htm	Applixware/Anyware and the Year 2000
http://www.aivazian.demon.co.uk/tt/tt.html	Time Travel - Linux Kernel Module for testing Year 2000 Compliance
http://www.sendmail.org/faq/section4.html#4.11	Sendmail Year 2000 Statement
http://www.sun.com/y2000/cpl.html	Sun Microsystems Year 2000 Product Compliance Status List
http://ca.samba.org/samba/docs/sambay2k.html	Samba Year 2000 Statement
http://www.scriptics.com/Y2K.html	Tcl Year 2000 Compliance
http://www.python.org/doc/FAQ.html#2.11	Python Y2K Statement
http://ourworld.compuserve.com/homepages/rsandler/Y2KFAQ.htm	Year 2000 FAQ
http://www.garynorth.com/	Y2K Links and Forums
http://www.ieeeusa.org/Y2K/	USA Y2K Resource Page

Altri problemi di data

Neppure il 1999 si può dire esente da spiacevoli sorprese legate alle date. Il 22 agosto ad esempio sono andati in crisi molti vecchi ricevitori GPS ed è stato necessario resettarli. Ciò a causa di un contatore interno che torna a zero ogni 1024 settimane (poco più di 19 anni).

Un'altra data fonte di problemi con cui dovremo fare presto i conti sarà il 9 settembre di quest'anno. Infatti in alcuni programmi la data 9/9/99 viene utilizzata come marcatore di errore o col significato di "mai". Quando si programma conviene essere lungimiranti (o almeno aver fiducia che il proprio lavoro duri per un po' di tempo).

Prima di tutto il BIOS

Non ha senso domandarsi se il sistema operativo o i programmi che si stanno utilizzando siano o meno compatibili con l'anno 2000 se prima non si è verificato che lo sia la macchina su cui stanno girando. Una fonte importante di problemi può essere infatti lo stesso firmware della macchina ed in particolare al BIOS dei PC.

Alcuni BIOS vecchi hanno infatti problemi con la gestione del cambio di secolo. Ciò non è un problema con altri sistemi operativi, in quanto essi stessi costringono gli utenti ad un upgrade abbastanza frequente della macchina. Linux invece funziona in modo egregio anche su macchine un po' vecchiotte e perciò deve fare i conti con questo problema.

La procedura per verificare se il proprio BIOS è compatibile con l'anno 2000 è abbastanza semplice e consiste nell'entrare nell'interfaccia di gestione dello stesso mediante la pressione dell'apposito tasto (DEL o ESC) al boot e nel regolare l'orologio a poco prima della mezzanotte fatidica, ad esempio alle ore 23:58 del 31 dicembre 1999. Spegnete ora il computer usando l'apposito pulsante di alimentazione (lo specifico perché certi computer invece di spegnersi si pongono in standby) e attendete 3 o 4 minuti prima di riaccenderlo. Se tutto è a posto, l'orologio dovrebbe segnare pochi minuti dopo la mezzanotte di sabato 1 gennaio 2000. Uno dei miei computer segna invece domenica primo gennaio 1900, che non è esattamente il risultato che mi sarei aspettato di ottenere.

Secondo la FAQ su <http://ourworld.compuserve.com/homepages/rsandler/Y2KFAQ.htm> ben 80.000.000 di PC nel mondo sarebbero affetti da questo problema.

Se anche il vostro BIOS è difettoso, è necessario che ve ne procuriate una versione più recente. L'aggiornamento, specialmente nel caso di macchine un po' vecchie, non è tuttavia dei più semplici, in quanto può essere necessario sostituire una EPROM nella scheda madre. I BIOS più recenti risiedono invece su Flash EPROM e perciò l'upgrade può essere effettuato da software (comunque essi non dovrebbero essere affetti da particolari problemi).

Il kernel di Linux ed il bug del 2000

Come regola generale, Linux **non** è afflitto dal bug del 2000.

Il kernel memorizza infatti le date contando i secondi trascorsi dalla mezzanotte del primo gennaio 1970 (data che prende il nome di "The Epoch", "L'Epoca"). In questo modo le ore 20:24:58 del 12 maggio 1999 vengono rappresentate col numero 926533498. Analogamente, la mezzanotte del 31 dicembre 1999 sarà per il kernel di Linux semplicemente un altro numero, senza un particolare significato (946677600). Vedremo in seguito che neppure questo metodo è esente da inconvenienti.

Per memorizzare questo valore la libreria ANSI C di UNIX utilizza la struttura "struct tm", definita in `/usr/include/time.h`, che dovrebbe essere manipolata solamente mediante alcune apposite funzioni standard del linguaggio (strftime) che permettono di ottenere o settare i diversi campi che compongono la data. Perciò i programmi che accedono correttamente a questa struttura non dovrebbero avere particolari problemi con l'anno 2000.

I programmi

Ma siamo sicuri che tutti i programmi funzionino correttamente? Provate a scrivere il seguente comando:

```
# date -s "Jan 1 00:00:00 2000"
```

Il risultato è un desolante:

```
date: invalid date `Jan 1 00:00:00 2000`
```

Il fatto che il kernel e le librerie standard non siano vulnerabili al problema non significa che lo siano automaticamente tutti i programmi che ci si fa girare sopra. Alcuni di essi potrebbero ad esempio accettare in input una data dall'utente e compiere essi stessi delle operazioni non compatibili con l'anno 2000.

Un errore di programmazione abbastanza frequente deriva dal fatto che nella "struct tm" il campo `tm_year` non rappresenta le ultime due cifre dell'anno, bensì "il numero di anni trascorsi a partire dal 1900".

```
struct tm
{
    int tm_sec; /* seconds */
    int tm_min; /* minutes */
    int tm_hour; /* hours */
    int tm_mday; /* day of the month */
    ...
    int tm_year; /* year since 1900 */
    ...
};
```

Il tipo di dati utilizzato (`int`) è abbastanza ampio da permettere un funzionamento corretto per almeno altri 32.768 anni, tuttavia la definizione del campo incoraggia i programmatori ad avere un modo apparentemente semplice per ottenere il valore delle ultime due cifre dell'anno. Se tale metodo funziona correttamente fino al 1999 (in cui `tm_year` vale 99), lo stesso non si può dire per l'anno seguente, in quanto, coerentemente alla sua definizione, `tm_year` non vale 00, bensì 100. Perciò è facile che alcuni programmi scrivano la data della prossima Epifania come 06/01/100 (se non addirittura 06/01/19100).

In generale i programmi di Linux più importanti e di uso più frequente non hanno particolare necessità di manipolare le date e se lo fanno passano comunque attraverso le funzioni messe a disposizione del sistema. Quasi tutti i comandi base di Linux derivano dal progetto GNU e già questo è un ottimo indice di affidabilità, robustezza e professionalità. Lo stesso dicasi per altro software molto utilizzato o scritto da organizzazioni importanti, come Apache, X Window o XFree. Come al solito la filosofia dell'Open Software aiuta moltissimo, in quanto ognuno è in grado di andare ad analizzare il codice sorgente dei programmi alla ricerca di

eventuali problemi ed eventualmente di risolverli.

Il sito <http://www.gnu.org/software/year2000-list.html> contiene la lista del software GNU di cui è stata testata la compatibilità con il bug del 2000. Per ogni programma (attenzione alle versioni!) è descritto se esso funzionerà correttamente il prossimo anno oppure se vi saranno dei problemi. Tali risultati non sono stati ottenuti mediante test banali come il portare avanti la data dell'orologio, che possono non prevedere tutti i casi possibili di errore, bensì da una analisi riga per riga del codice sorgente, il che offre buone garanzie di correttezza.

In certi casi troverete scritto "non siamo sicuri che funzioni". Nonostante la bravura delle persone che scrivono software sotto l'etichetta GNU capita infatti che certi programmi siano scritti in un linguaggio che assomiglia più all'aramaico antico che al C moderno. In questo caso fare una analisi riga per riga risulta abbastanza arduo. A questo proposito mi viene in mente una gara che avviene ogni anno fra i programmatori C, in cui vince chi scrive il programma più illeggibile (non a caso il nome della gara è "Obfuscated C Contest"). Il regolamento impone tuttavia che non solo tale codice sia formalmente corretto e compilabile mediante un compilatore ANSI C, ma anche che esso possa essere passato con successo attraverso lint, un tool che ne verifica ulteriormente ed in modo assai severo la correttezza.

Anche all'interno del software GNU vi sono tuttavia dei programmi non compatibili con l'anno 2000: alcuni di questi infatti si basano moltissimo sulla manipolazione delle date, quelli che gestiscono le diverse revisioni del software, come CVS.

È bene ripetere che è necessario verificare con una certa accuratezza la versione dei programmi che si stanno utilizzando. Non è ad esempio "Apache" ad essere compatibile con l'anno 2000, bensì la "versione 1.3 di Apache". La differenza non è da poco.

Sebbene i programmi importanti siano per la quasi totalità immuni da problemi dovuti all'anno 2000 o perlomeno essi sono almeno conosciuti, lo stesso non si può dire per tutti quei programmi piccolissimi e poco utilizzati, in quanto spesso nessuno, compresi gli stessi autori, si è preso la briga di andare a ve-

rificarne ed eventualmente correggerne i problemi. Poco male: se nessuno se ne è preoccupato, è probabile che esso non compia operazioni particolarmente critiche o irrinunciabili.

La migliore fonte di informazioni sulla compatibilità di un programma con l'anno 2000 rimane quella di andarsi a leggere la documentazione fornita con esso oppure di visitarne la homepage su Internet. Alcuni indirizzi utili sono riportati nei bookmark nel riquadro allegato a questo articolo.

Nel caso del software commerciale, spesso non si può disporre dei sorgenti. In questo caso la verifica della presenza del bug e le eventuali correzioni possono essere eseguiti solamente dagli autori. E se la ditta è fallita? Pazienza, è uno dei difetti del software commerciale.

Time Travel

Come ultima risorsa esiste Time Travel (<http://www.aivazian.demon.co.uk/tt/tt.html>), un piccolo modulo (2111 byte) da caricare al boot, il quale permette di testare un singolo programma con una data diversa da quella del resto del sistema. Oltre a permettere le prove di compatibilità, esso sarà utile il prossimo anno per far girare eventuali programmi bacati come se fossero ancora nel 1999. Ovviamente l'autore si raccomanda di utilizzarlo solamente a questo scopo e non per aggirare le protezioni dei programmi.

Il funzionamento di Time Travel è semplice: esso intercetta le chiamate al sistema operativo che hanno a che fare con la data e simula un ambiente con un valore diverso dell'orologio. L'installazione è semplicissima, in quanto non richiede la ricompilazione del kernel. Una volta ottenuto il programma lo si carica in memoria con un semplice

```
# insmod timetravel.o tt_prog="myprog" tt_shift=100000 Kevin Schick, del Gartner
```

dove al posto di myprog si deve sostituire il nome del programma da testare. Il valore di tt_shift rappresenta il numero di secondi che si vogliono aggiungere o togliere alla data attuale.

Il 19 gennaio 2038

E veniamo al problema di data che affligge la maggior parte dei sistemi

UNIX ed anche Linux. Il metodo inventato da Ritchie e colleghi di rappresentare le date come i secondi trascorsi dall'Epoca, se applicato ad un sistema a 32 bit è in grado di contare solamente 2^{32} secondi. In pratica il contatore viene azzerato dopo che sono passati 2.147.483.647 secondi, che è il massimo numero che può essere rappresentato mediante un intero a 32 bit con segno. Anche se a prima vista possono sembrare molti, in pratica causano l'overflow del registro che contiene la data più o meno all'inizio del 2038. Come passa in fretta il tempo...

Il problema può essere risolto o riscrivendo in modo complicato parti del kernel, oppure passando ad un sistema a 64 bit, che consente di memorizzare date fino all'anno 292.271.025.015.

Anche se quasi quarant'anni di tempo possono sembrare molti, sono sicuro che nel 2037 ci sarà sicuramente qualcuno che si troverà a dover affrontare in fretta e furia il problema. D'altronde anche il problema del 2000 è conosciuto da parecchi anni e nonostante questo c'è chi ha continuato imperterrito a scrivere programmi che considerano solo le ultime due cifre della data.

Conclusioni

Più che il dover fare l'inventario delle applicazioni non funzionanti ed eventualmente correggerle o aggiornarle alle versioni corrette, l'aspetto che maggiormente preoccupa è il poco interesse da parte di molte aziende a dedicare soldi o risorse in tale operazione. Si tratta infatti di un impegno assai gravoso e a cui le aziende spesso non sono preparate, vuoi per la mancanza di personale preparato, vuoi per il fatto che, in quanto non retribuito, viene considerato un lavoro non prioritario. Secondo Kevin Schick, del Gartner Group, "l'alternativa all'affrontare per tempo il problema del 2000 sarà il fallimento dell'azienda".

PS: Il primo gennaio cade di sabato e perciò ci sono eventualmente ben due giorni per cercare di procurarsi una nuova radiosveglia nel caso quella vecchia non funzioni. Peccato, perché come alibi non era niente male.