

Fixing the year 2038 problem

FOSDEM 2023

Wookey

wookey@wookware.org

The problem

- Time (time_t) on unix 32-bit systems is 32-bit signed int seconds since 1st Jan 1970. Rolls over at 2147483647 to negative number.
- Happens at 2038-01-19 03:14:07 UTC. Next second is 1901-12-13 20:45:52 UTC
- This is slightly less than 15 years away
- There are other issues: Y2106 (unsigned 32-bit Epoch-based times), Y2107 (FAT/vfat), Y2036 (unsigned 32-bit 1900-based RFC 868 times (NTP)) or Y9999 (four-digit years)

This session

- I am not much of an 'expert'.
- Explain the problem
- Collect feedback/issues/things to check
- Test level of consensus
- Hopefully end with some sort of plan for a plan.

How big a problem?

- Real computing mostly 64 bit already
- Lots of cheap computing still 32 bit
- Cars, TVs, controllers (buildings, plant, IOT), cheap phones
- A lot of stuff still running in 2038 is already installed
- There will still be new things. We should fix them. Soon.
- A lot more OpenEmbedded/Android than binary distros.
- Debian has bigger problem than most

Who cares?

- riscv32, x32, loong32, arc already 64-bit time
- RHEL already dropped 32-bit arches
- armv5, i386, mips32 and ppc32 already close to obsolete
- Source distros 'just' rebuild:
openwrt/buildroot/yocto/gentoo/adelie
- Leaving debian armv7 ('armhf'). Anyone else?

Work so far

- Arnd Bergman and Deepa Dinamani – Kernel, 2017.
<https://lwn.net/Articles/717076/>
- Perl fixed 5.12 (works on 32-bit system support) (2010)
- Musl is fixed in 1.2 (2020): <http://musl.libc.org/time64.html>
(time_t always 64bit, but things can still run using old ABI)
- Glibc fixed in 3.34 (2021)
- Lots of other software fixed – how much is still broken?

Distro work

- Adelle (musl):
<https://web.archive.org/web/20220301175235/https://wiki.adelielinux.org/wiki/Project:Time64>
- Gentoo: https://wiki.gentoo.org/wiki/Project:Toolchain/time64_migration
- OpenSuse: https://www.reddit.com/r/linux/comments/xjtf3q/in_the_year_2038/
- Ubuntu: Library analysis https://people.canonical.com/~vorlon/armhf-time_t/
- OpenEmbedded: Assorted patches
- Debian: Rebuild attempt in 2020 (too much broken)
Rebootstrap base (2022), ABI analysis

Glibc

- Glibc 3.34 – supports old and new.
- Does not force 64bit so other stuff just rebuilds
- File format changes: utmp, wtmp, btmp have a time_t in
- Qemu usermode (32 on 64) bust in 2.37 (patches exist)
- Sets FILE_OFFSET_BITS=64 if TIME_BITS=64

LFS (Large Filesystem Support) is involved

- Glibc 3.34 enforces `_FILE_OFFSET_BITS=64` if `_TIME_BITS=64`
- `_LARGEFILE_SOURCE` enables both 32 and 64bit ABI
- `_FILE_OFFSET_BITS=64` enables 64bit ABI only.
- LFS is similar transition to 64bit `time_t` in this regard

Gnulib and autoconf

- gnulib automatically enables time64 support if the system supports it.
 - set `gl_cv_type_time_t_bits_macro=no` to stop it
- Autoconf 2.72 (Nov 2022) release tried to tie LFS and 64bit-time. (Reverted).

ABIs and files

- ABI changes if `time_t` used in struct
- File and disk formats contain 32-bit times
- New ABI is just like any other ABI bump, but HUGE
- Supporting/transitioning old file formats important for apps

Fundamental question

- **Update existing architecture? (arm-linux-gnueabihf)**
 - Most efforts so far
 - Easy for source distros
 - Significant risk of breakage for binary distros
 - Changes the ABI
- **New ABI → new triplet?**
 - More 'correct'?
 - Easier for binary distros? (Some stuff will break with new names)
- **Some consensus would be good**

Debian migration

- Just rebuilding against newer glibc doesn't cause transition something has to set `_TIME_BITS=64` and `_FILE_OFFSET_BITS=64`
- Dpkg, glibc or gcc could set them (glibc best?).
- New arch/triplet is simpler ('arm32'/arm_linux_gnuxxxx)
- Big transition like libc5→libc6 in existing arch/triplet Process exists but blocks migration on all arches.
- Minor-arch ABI transition example: long double changed from 64-128bits on alpha, powerpc, sparc, s390 (2007)

How big a problem 2?

- 6429 packages of Debian's 35960 have time_t in
- How many in public ABIs and file formats?
- 7 of bottom 85 libs have changed ABI
- Of 767 library packages: 209 analyser failed, 558 checked
82 changed ABI, 476 did not. (17%) (ubuntu tests)
So maybe 113 libs in transition?
- Mixing binaries worked fine in my tests so far.

What else?

- NFSv3 (might be signed, might not), ext3, XFS, cpio?
- INN has `time_t` in file format
- cpio uses 33bit (11 octal digits) for mtime. OK till 2174?
- 32-bit Wine
- What else will break? (in 2038?/on upgrade now?)

Questions

- What doesn't build with 64bit `time_t`? Or LFS?
- Non-ABI issues?
- Things we should test?
- New triplet/arch or transition-in-place?
- Glibc - flag day/triplet/libdir/something else?
- Timescale?

Discussion

- distributions@lists.linux.dev
- <https://subspace.kernel.org/lists.linux.dev.html>
- Please join in if you are interested/worried/understand specific issues.

Modern C porting also implicated

- Implicit function defines are going away
- Errors from gcc14 and clang 16 (next year)
- https://wiki.gentoo.org/wiki/Modern_C_porting
- Implicit functions break glibc foo→foo64 macros
- Some configure tests depend on implicit functions

Tools

- Abigail-tools
- Abi-dumper
- ABI-compliance-checker:
<https://lvc.github.io/abi-compliance-checker/>
- More?