Debian 13 trixieで変わった 64bit time_t 移行について

MiniDebConf Japan 2025

Norimitsu SUGIMOTO (杉本 典充)
dictoss@live.jp

2025-09-06

この資料について

- 東京エリア Debian 勉強会の Web サイトから PDF をダウンロードできます
- https://tokyodebian-team.pages.debian.net/2025-09-minidebconf.html
- 勉強会は関西 Debian 勉強会と一緒にオンラインでほぼ毎月開催しています

目次

自己紹介

- Norimitsu SUGIMOTO (杉本 典充) dictoss@live.jp
- Twitter: @dictoss
- MiniDebConf Japan 2025 の実行委員長
- Debian JP Project 会長 (2024-2025)
- 北海道旭川市の出身
- 北海道旭川北高等学校、北海道教育大学旭川校を卒業
- 東京でシステム開発の仕事をしています (python などで Web サイト開発、通信プログラム開発)

2038年
問題とは

2038年問題とは

- UNIX 時間という「1970年1月1日 0時0分0秒」を0として1秒ごとに1加算する時間の表現方法
- UNIX/Linux の多くではUNIX時間を表現する time_t 型を用いており、32bit OS の場合は「long 型」という4バイト符号付き整数を用いていた
  - 4バイト符号付き整数の場合はUTCで 2038年1月19日3時14分7秒を過ぎるとオーバーフローする。**数値が 0 になり 1970 年に戻ってしまうことで**コンピュータが誤作動する可能性がある
  - このことは**「2038 年問題」**と呼ばれている
- UNIX/Linux の多くの 64bit OS の場合
  - long 型は 8 バイト符号付き整数になるため、2038 年問題は発生しない

Debian 13 trixie の 2038 年問題対応 (1)

- リリースノート 2.2.6. 64 ビット time_t ABI への移行[1]
  - i386 を除くすべてのアーキテクチャは **64 ビット time_t ABI** を使うようになり、2038 年を超えた日付をサポートするようになりました。
  - 32 ビットのアーキテクチャ (armel および armhf) では、多数のライブラリの ABI 変更がそのライブラリの "soname" の変更なしに実施されました。これらのアーキテクチャでは、サードパーティーのソフトウェアやパッケージは**再コンパイル・再ビルドが必要**で、そして静かにデータ損失している可能性があるので精査が必要です。
  - **i386 アーキテクチャはこの移行の対象外**です。これは第一の役割がレガシーなソフトウェアをサポートすることだからです。

[1] https://www.debian.org/releases/trixie/release-notes/whats-new.ja.html#bit-time-t-abi-transition

Debian 13 trixie の 2038 年問題対応 (2)

なぜ trixie のタイミングなのか

- 2020-03-29 linux-5.6 リリース
  - 32bit 版でも時刻を 64bit 整数で保持するように変更
- 2021-08-01 glibc-2.34 がリリース
  - _TIME_BITS=64 マクロに対応

日付	linux	glibc	debian
2020-03-29	linux-5.6	-	-
2021-08-01	-	glibc-2.34	-
2021-08-15	-	-	debian 11 linux-5.10, glibc-2.31
2023-07-22	-	-	debian 12 linux-6.1, glibc-2.36
2025-08-09	-	-	debian 13 linux-6.12, glibc-2.41

trixie 向けパッケージで必要な対応

trixie 向けパッケージで必要な対応 (1)

- コンパイルオプションの変更
- debian/control の修正
- ビルドとインストール確認
- ソフトウェア自体の設定・実装の確認
- パッケージのアップロード (説明は割愛)

trixie 向けパッケージで必要な対応 (2)

パッケージのコンパイルオプションが変更
ビルドするホスト上で dpkg-buildflags コマンドを実行すると確認可能

```
$ sudo apt-get install build-essential devscripts
$ dpkg-buildflags
```

armhf の debian12 と debian13 における CPPFLAGS の差分

```
-CPPFLAGS=-Wdate-time -D_FORTIFY_SOURCE=2
+CPPFLAGS=-D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64 -D_TIME_BITS=64
   -Wdate-time -D_FORTIFY_SOURCE=2
+CPPFLAGS_FOR_BUILD=-D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
  -D_TIME_BITS=64 -Wdate-time -D_FORTIFY_SOURCE=2
```

2GB 超えファイル対応と _TIME_BITS=64 マクロが追加

trixie 向けパッケージで必要な対応 (3)

debian/control の修正
例示）libmyapp を移行する場合

- Package: libmyappt64
  - 提供するライブラリパッケージ名の末尾に t64 を追加する命名規則になった
- Replaces: libmyapp
  - bookworm からのアップグレード時に従来のパッケージ名から t64 が末尾につくよう変更したことを示す
- Build-Depends: dpkg-dev (>= 1.22.5)[2]
  - -D_TIME_BITS=64 を指定するようになった初めての dpkg-dev のバージョン
- Depends: (依存するライブラリで t64 が末尾にあるパッケージは全て修正)

[2] https://wiki.debian.org/ReleaseGoals/64bit-time

trixie 向けパッケージで必要な対応 (4)

ビルドとインストール確認

```
$ debuild -uc -us
```

パッケージのクリーンビルド

- pbuilder / cowbuilder
- sbuild

インストールテスト

- autopkgtest
- piuparts

ソフトウェア自体の設計・実装の確認

ソフトウェア自体の設計・実装の確認 (1a)

```
$ vi src/time64.c

#include <stdio.h>
#include <time.h>

struct mytime{
    int a;    /* 4 byte */
    time_t t; /* 4-8 byte */
    int b;    /* 4 byte */
};

int main(int argc, char *argv[]){
    printf("sizeof(int)    = %lu\n", sizeof(int));
    printf("sizeof(long)   = %lu\n", sizeof(long));
    printf("sizeof(time_t) = %lu\n", sizeof(time_t));
    printf("sizeof(struct mytime) = %lu\n", sizeof(struct mytime));
    return 0;
}
```

```
$ make
gcc -Wall -g -c src/time64.c -o src/time64.o
gcc -Wall -g -o a.out  src/time64.o
```

ソフトウェア自体の設定・実装の確認 (1b)

環境による変数のバイト数の違い

バイト数	Debian 12 amd64	Debian 12 armhf	Debian 13 armhf
sizeof(int)	4	4	4
sizeof(long)	8	4	4
sizeof(time_t)	8	4	8
sizeof(struct mytime)	24	12	24

- time_t 型の変数のサイズ
  - 4 byte から 8 byte へ増える
- time_t 型を含む構造体のメモリサイズとオフセット
  - サイズが (パディングで) 大きくなる場合がある
  - 構造体のメモリをそのままバイナリでファイルへ書き込む場合や通信で送信している場合は旧システムと互換が無くなる

ソフトウェア自体の設計・実装の確認 (2)

古いバージョンの Debian から 64bit time_t 移行の注意点

- ファイルシステム
  - ext 系は ext4 へ移行すること (ext2/ext3 は 2038 年問題あり)
  - xfs は linux-5.10 より古い環境でフォーマットした場合は 2038 年問題あり
- データベース
  - 型定義で時刻を 32bit 数値で扱っている場合あり (例: mysql の timestamp 型)
- NTP サーバ / NTP クライアント
  - NTP (プロトコル) の 2036 年問題 (1900 年始まり)
    - 2010 年発行の RFC5905 を実装した NTPv4 対応版なら影響なし
    - Debian 12 から NTP クライアントのデフォルトは systemd-timesyncd
    - Debian 12 で ntp パッケージは移行パッケージになり ntpsec へ置換
    - その他の移行先は chrony、openntpd が選択肢

まとめ

- Debian 13 trixie では debian/control の作法が変わった
- 32bit 環境では time_t 型は debian パッケージのビルド時以外でも標準で 64bit へ変更 (i386 は除く)
- time_t 型のサイズ変更で構造体のメモリのサイズやオフセットが変わっても、プログラムが扱うファイルや通信で互換性があるか確認が必要
- プログラムを再ビルドする以外にも 32bit time_t 問題が隠れており残さず対応が必要

参考文献

- Debian wiki - ReleaseGoals 64bit-time
  https://wiki.debian.org/ReleaseGoals/64bit-time
- Yocto よもやま話 第 3 回「Linux の 2038 年問題」
  https://www.lineo.co.jp/blog/yocto/vol03-2038.html