# .Deb

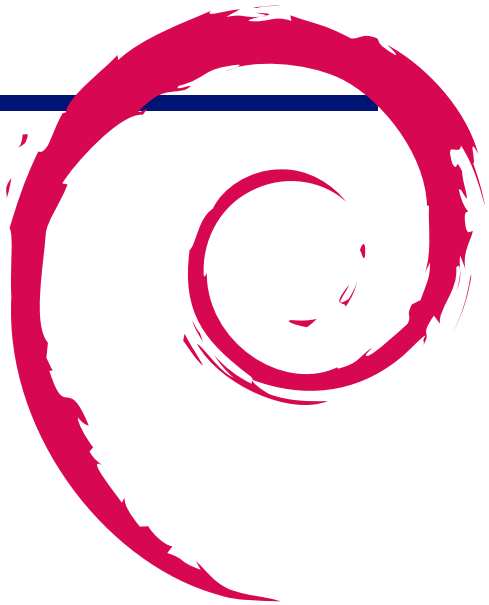
銀河系唯一のDebian専門誌

2023 年 8 月 19 日

## ２０３８年問題対策特集

# デビアン勉強会

## 目次



# 1 最近の Debian 関連のミーティング報告

杉本 典充

## 1.1 2023 年 7 月度 東京エリア・関西合同 Debian 勉強会

2023 年 7 月 15 日 (土) に東京エリア Debian 勉強会と関西 Debian 勉強会の合同でオンラインによる Debian 勉強会を開催しました。参加者は 9 名でした。

セミナー発表は、林 さんによる「Debian Installer の日本語フォントをどうにかする話」を行いました。

勉強会の終了後、参加者同士で Debian や OSS に関する話の情報交換を行いました。

# 2 事前課題

杉本 典充

今回の事前課題は以下です。

1. Debian に Live インストールイメージがあることを知っていますか。
2. Debian の 32bit 版を使っていますか (Debian の派生 OS を使っている場合は使っている扱いでお答えください)。

## 2.1 dictoss

1. 知っているが、使ったことはない
2. armel / armhf を使っている

## 2.2 NOKUBI Takatsugu (knok)

1. 知っており、Debian 11 版以前を使ったことがある
2. i386 を使っている

## 2.3 Hiroyuki Yamamoto (yama1066)

1. 知っており、Debian 11 版以前を使ったことがある
2. 使っていない

## 2.4 takaswie

1. 知っているが、使ったことはない
2. i386 を使っている

## 2.5 kenhys

1. 知っているが、使ったことはない
2. armel / armhf を使っている

## 2.6 yy_y_ja_jp

1. 知っており、Debian 11 版以前を使ったことがある
2. i386 を使っている

## 2.7 あおき (sambar_gamer)

1. 知っており、Debian 11 版以前を使ったことがある
2. armel / armhf を使っている

## 2.8 tMatsu

1. 知らなかった
2. armel / armhf を使っている

## 2.9 hihitani

1. 知っているが、使ったことはない
2. i386 を使っている

## 2.10 カナやん

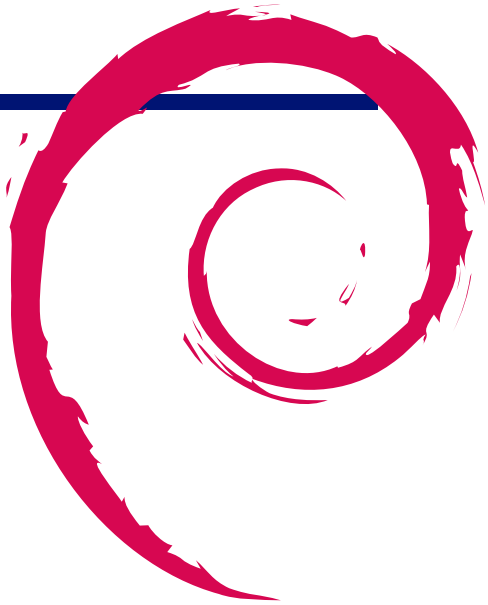
1. (回答なし)
2. (回答なし)

## 2.11 ysaito

1. 知らなかった
2. armel / armhf を使っている

## 2.12 jsynth21

1. 知っており、Debian 11 版以前を使ったことがある
2. 使っていない



# 3 2038 年問題と 32bit OS の 64bit time_t への移行について

杉本 典充

## 3.1 はじめに

2023 年 5 月頃に debian-devel@lists.debian.org へ Debian の 32bit 版アーキテクチャにおける time_t 型を 32bit 長から 64bit 長 に変更してはどうかという提案のメールが投稿されました[*1]。

コンピュータの世界には「2038 年問題」という言葉があり、メールにあるこの提案は過去のソフトウェア資産が多い i386 版と 64bit CPU をあまり必要としない組み込み系でよく使われるアーキテクチャにおいて避けては通れない対応です。

次の Debian 13 trixie でこの提案が適用されるかはまだ議論中ですが、変更することが決まった場合に debian のパッケージメンテナンスを行う立場においてどのような考慮が必要で、どのような対応を行うことになるのか調べてみました。

## 3.2 2038 年問題と コンピュータのデータモデルの関係

### 3.2.1 2038 年問題とは

「2038 年問題 (the year 2038 problem)」とは、”UNIX 時間” が 1970/01/01 00:00:00 (UTC) を起点に 1 秒ごとにカウントアップする数値で表現する仕様にしているため、符号付き 32bit 整数で表現している場合に 2038/01/19 03:14:07 (UTC) を過ぎるとオーバーフローする問題です。

UNIX はベル研究所で 1969 年で開発されており[*2]、その後 32bit CPU が主流になってからは UNIX 時間を符号付き 32bit 整数 (signed long) で実装する UNIX/Linux 系 OS が多く見られました。UNIX/Linux 系 OS の 多くの 32bit OS は 2023 年現在でもこの仕様で動き続けているため、あと 15 年 で UNIX 時間がオーバーフローするところまで来てしまっています。

### 3.2.2 32bit OS と 64bit OS におけるデータモデルの違い

コンピュータにおけるプログラミングでは、”変数” というデータをメモリ上で読み書きする領域を宣言します。このとき、C 言語 においては変数の名前とともには ”データ型” を宣言します。

OS ごとにデータ型の名称とメモリサイズを定義をしており、そのセットを ”データモデル” と言います。UNIX/Linux 系の 32bit OS の多くでは「ILP32 モデル」、64bit OS の多くでは 「LP64 モデル」というデータモデルが採用されています[*3]。

[*1] https://lists.debian.org/debian-devel/2023/05/msg00168.html

[*2] https://museum.ipsj.or.jp/computer/unix/history.html

[*3] 64bit OS 向けのデータモデルは他に「LLP64 モデル」「ILP64 モデル」がありますが採用例は多くありません。

「ILP32 モデル」と「LP64 モデル」におけるメモリサイズの違いを 表 1 に示します。

表 1　データモデルによるデータ型の名称とメモリサイズの違い

| データ型 | ILP32 モデル | LP64 モデル |
|---|---|---|
| char 型 | 1 バイト | 1 バイト |
| short 型 | 2 バイト | 2 バイト |
| int 型 | **4 バイト** | **4 バイト** |
| long 型 | **4 バイト** | **8 バイト** |
| long long 型 | **8 バイト** | **8 バイト** |
| float 型 | 4 バイト | 4 バイト |
| double 型 | 8 バイト | 8 バイト |
| ポインタ型 | **4 バイト** | **8 バイト** |

「ILP32 モデル」と「LP64 モデル」のデータモデルを比較すると、表 1 の下線の部分に特徴が見られます。

1. int 型のサイズは同じ (int 型は普通「CPU のレジスタの自然長の大きさ」であるが、LP64 モデルでは 4 バイト と短い)
2. long 型のサイズは異なる
3. long long 型のサイズは同じ
4. ポインタ 型は「CPU のレジスタの自然長の大きさ」でサイズは同じ

C 言語では時刻を表現するデータ型として「time_t 型」が定義されており <time.h> を include すると使えます。time_t 型は Debian を含む UNIX/Linux 系 OS の多くでは singed long 型で定義されているため 2. が適用され、32bit OS と 64bit OS で表現できる UNIX 時間の上限に差が出ます。つまり、2038 年問題は 64bit OS では発生せず、32bit OS のみで発生します。

## 3.3　OSS における 2038 年問題への対応

### 3.3.1　linux-5.6 における UNIX 時間の 64bit 化

Linux カーネルにおいては、2020 年 3 月 29 日にリリースした linux-5.6 で 32bit 版でもカーネル内部の UNIX 時間を符号付き 64bit 整数で処理するよう改修が行われました。この改修でユーザ空間からアクセスした場合に UNIX 時間をそのまま 64bit time_t 型の値で返すだけではユーザ空間のソフトウェアに影響が出ます。そのため、ユーザ空間が 32bit なアプリケーションとの互換性を持たせる CONFIG_COMPAT_32BIT_TIME[*4] という設定が kernel config に追加されました。この設定を y にした Linux カーネルではユーザ空間からアクセスするインタフェースを 64bit time_t 型と 32bit time_t 型の両方をサポートします[*5]。

ただし、これらの対応は Linux カーネル内部の話のため、ファイルシステムにおいてはタイムスタンプのサイズが 32bit である ext2、ext3、ReiserFS では UNIX 時間がオーバーフローする問題は残ったままになります[*6]。

### 3.3.2　glibc-2.34 における対応

glibc においては、2021 年 8 月 1 日にリリースした glibc-2.34[*7] で time_t 型を扱う関数が 32bit time_t 型を扱う関数と 64bit time_t 型を扱う関数の両方を持つようになりました。glibc-2.34 の 32bit 版における time_t 型のデフォルトサイズはこれまで通り 32bit のままですが、glibc をリンクするソフトウェアのビルド時に _FILE_OFFSET_BITS=64 および _TIME_BITS=64 のマクロを両方定義すると time_t 型と時間処理の関数は

[*4] https://cateee.net/lkddb/web-lkddb/COMPAT_32BIT_TIME.html
[*5] Debian 12 bookwork の i386、amd64 では CONFIG_COMPAT_32BIT_TIME=y でビルドされています。
[*6] ext4 のタイムスタンプの上限は 2514 年までのため、ext4 へ移行する対応策があります。
[*7] https://sourceware.org/pipermail/libc-alpha/2021-August/129718.html

64bit time_t 型を使うように切り替えることができます。

この glibc の実装方法では time_t 型のサイズはユーザ空間で動作するソフトウェアをビルドするときに決定されるため、32bit OS で 2038 年問題を解決するにはユーザ空間のソフトウェアをすべて再ビルドする必要があります。

### 3.3.3 OS 毎、ディストリビューション毎の対応状況

参考までに世の中の OS はどのような対応状況になっているかを 表 2 にまとめてみました。

表 2　表 OS ごとの 64bit time_t の対応状況

| OS 名 | 32bit 版 | 64bit 版 |
|---|---|---|
| Debian | **未対応** | 対応済み |
| Ubuntu | 19.10 から提供なし | 対応済み |
| Red hat Enterprise Linux | 2014 年リリースの 7 から提供なし | 対応済み |
| NetBSD | 2012 年リリースの 6.0 から 64bit 化 | 対応済み |
| OpenBSD | 2014 年リリースの 5.5 から 64bit 化 | 対応済み |
| FreeBSD | **未対応** | 対応済み |
| Windows | 2021 年リリースの 11 から提供なし | 対応済み |

## 3.4 Debian における 2038 年問題への対応案

### 3.4.1 Debian における 2038 年問題へ対応するための議論

debian においても 2038 年問題への対応について議論が行われてきました。

DebConf 17 で行われた BoF[*8] では、まず kernel や glibc の対応がなされないとディストリビューション側で独自に対応するのは困難という意見でした。その後 2020 年に linux-5.6 がリリース、2021 年に glibc-2.34 がリリースされ、32bit OS で 64bit time_t 型を扱えるようになりました。

Debian 12 bookworm は 2023 年 7 月 22 日にリリースしており、linux-6.1 と glibc-2.36 を採用しています (Debian 11 bullseye は linux-5.10 と glib-2.31 を採用) 。つまり、Debian 12 bookworm でようやく 32bit OS で 64bit time_t 型 を扱える linux カーネルと glibc の対応したバージョンを両方クリアしました。

そのため、そろそろ debian も 32bit OS における 2038 年問題を解決するために行動を起こすべきだという提案のメール[*9] が 2023 年 5 月 16 日に投稿されたという経緯になっています。この提案については、Debain Wiki[*10] に背景と技術的な解説の詳細が書いてあります。

ただこの提案を実行に移すには以下の難点があります。そのため、適用される 32bit の CPU アーキテクチャは過去のソフトウェア資産の量を考慮して、切り替えるのかまたは新しいアーキテクチャをつくるのかが検討されるものと思います。

- ABI/API が変わるため、過去にビルドしたバイナリパッケージとの互換がなくなり古いアプリケーションやライブラリが動かなくなる
- C 言語で書かれたパッケージのうち、time_t 型を使っているパッケージをすべて再ビルドする必要がある

### 3.4.2 パッケージのメンテナが 64bit time_t 型に対応する技術的な話

**提案中: future=+time64 オプションの追加**

[*8] https://lists.debian.org/debian-devel/2017/09/msg00035.html
[*9] https://lists.debian.org/debian-devel/2023/05/msg00168.html
[*10] https://wiki.debian.org/ReleaseGoals/64bit-time

前述のとおり、64bit time_t 型に対応するにはかなりの数のパッケージを再ビルドをする必要があります。そのため、debian パッケージのビルドシステムの中で対応できるオプションを追加する案があり、これは環境変数 DEB_BUILD_OPTIONS および DEB_BUILD_MAINT_OPTIONS (man dpkg-buildflags 参照) で指定できる ”future=+time64” というもので、すでにバグ報告がされておりパッチも出ています[*11]。

このバグ報告を見ると future=+time64 はデフォルトでは指定されないような形で導入される模様です。そのため ”future=+time64” のオプションとすでに存在する ”future=+lfs”[*12] の両方を、パッケージ固有のビルドオプションを追加する環境変数 DEB_BUILD_MAINT_OPTIONS に定義するよう debian/rules を修正すると対応完了となります。

```
export DEB_BUILD_MAINT_OPTIONS = future=+lfs future=+time64
```

この設定をした debian/rules を使って debian パッケージをビルドすると-D_FILE_OFFSET_BITS=64 -D_TIME_BITS=64 をコンパイルオプションに付与した状態でソースコードをコンパイルできます。

なお future=+lfs future=+time64 オプションは 64bit 環境でのビルドでは効果がないため、CPU アーキテクチャごとの分岐を作りこむ必要はありません。

#### future=+time64 オプションを待てない方向けの方法

とにかく早く自分のパッケージを 64bit time_t 型に対応させたい場合は、-D_TIME_BITS=64 と _FILE_OFFSET_BITS=64 をコンパイルオプションに含めて debian パッケージをビルドすればよいです。

すでに独自に 64bit time_t 型の対応が入っているパッケージに Debian 12 bookworm の tar パッケージ (1.34+dfsg-1.2) があり、tar パッケージの debian/rules は以下のように書かれています。

```
#!/usr/bin/make -f

include /usr/share/dpkg/architecture.mk

(中略)

export DEB_BUILD_MAINT_OPTIONS = future=+lfs
export DEB_CFLAGS_MAINT_APPEND = -Wall -Wno-analyzer-null-argument
# Workaround gnulib issue: The below three lines can be dropped once
# tar >= 1.35 is used.
ifeq (32,$(DEB_HOST_ARCH_BITS))
export DEB_CPPFLAGS_MAINT_APPEND = -D_TIME_BITS=64
endif

(以下、省略)
```

重要な点は以下の 2 行です。

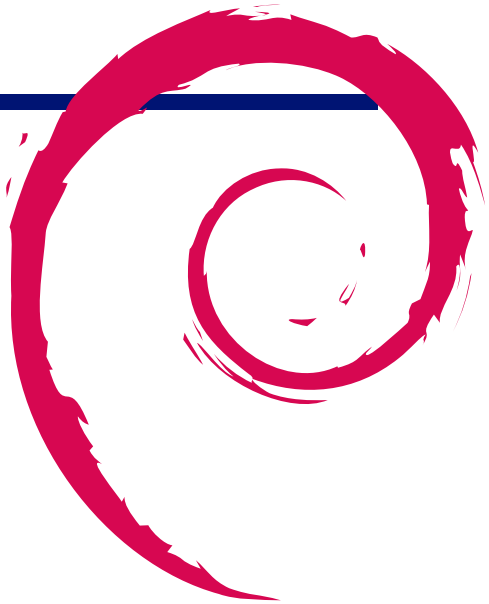
- export DEB_BUILD_MAINT_OPTIONS = future=+lfs
- export DEB_CPPFLAGS_MAINT_APPEND = -D_TIME_BITS=64

DEB_CPPFLAGS_MAINT_APPEND は ソースコードのコンパイルオプションである CPPFLAGS にパッケージ固有のビルドオプションを追加する環境変数です (man debhelper 参照)。なお CPPFLAGS の部分は CFLAGS、CXXFLAGS、LDFLAGS などに変えた環境変数も使えるため、ビルドするソフトウェアに合わせたものを使うとよいでしょう。

## 3.5 終わりに

32bit 版を今後も必要とする方向けに debian で 2038 年問題に対応する方法を紹介しました。将来長くに渡って動作し続ける debian を提供できるようがんばっていきましょう。

[*11] https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=1030159
[*12] Large-file support の意味で、32bit 版で 2 GB を超えるファイルを扱えるようになります。



# 4 メモ

**Debian 勉強会資料**

2023 年 8 月 19 日　　初版第 1 刷発行

東京エリア Debian 勉強会 （編集・印刷・発行）