(帰ってきた)
より新しいMozcを
Debianでも使いたい

Bazel (/'beɪz¸əl/)とのつきあいかた

Kentaro Hayashi

ClearCode Inc.

2026年08月 東京エリア・関西合同Debian勉強会

スライドは
Rabbit Slide Showにて公開済み

- (帰ってきた)より新しいMozcをDebianでも使いたい
  - https://slide.rabbit-shocker.org/authors/kenhys/tokyodebian-update-mozc-202608
    - 2024年11月のDebian勉強会「より新しいMozcをDebianでも使いたい」という発表の改訂版

本日の内容

- Debianにおける日本語入力関連パッケージ
- UpstreamであるMozcの動向
- DebianにおけるMozcのメンテナンス状況
- DebianのMozcを更新する試み
  - 更新の壁/どこまで視野にいれるか
  - Bazel対応の7つのポイント
  - 現在の状況と残作業

前回(2024/11)からの主要な差分

- 👍 Bazel 7.7.1がunstableで利用可能になった
- 👍 Mozc **3.33.6133**のPoCをexperimentalに投入済み
  - 👎 将来的にMozcの32bit環境(i386, armhf)サポートは終了します
- 🤔 uim-mozcはまだ必要ですか？

本日の内容

- **Debianにおける日本語入力関連パッケージ**
- UpstreamであるMozcの動向
- DebianにおけるMozcのメンテナンス状況
- DebianのMozcを更新する試み

Debianにおける日本語入力関連パッケージ

- task-japanese-desktop
  - 日本語を扱うデスクトップ環境にインストールすべきものを規定
    - 推奨はMozcもしくはAnthy
    - **uim-mozc,uim-anthyは推奨から除外** (See tasksel 3.87)

      (revertされなければ) Debian 14(forky)から

task-japanese-desktop

```
$ apt depends task-japanese-desktop
task-japanese-desktop
  依存: tasksel (= 3.88)
 |推奨: firefox-esr-l10n-ja
  推奨: firefox-l10n-ja
  推奨: fonts-vlgothic
  推奨: fonts-ipafont
 |推奨: ibus-mozc
  推奨: ibus-anthy
  推奨: mozc-utils-gui
 ...
```

GNOME環境だったら？

- task-japanese-gnome-desktop
  - 日本語を扱う**GNOMEデスクトップ環境**の推奨を規定
    - 推奨はibus-mozcもしくはibus-anthy (変更なし)
    - 通常ibus-mozcが選択される

KDE環境だったら？

- task-japanese-kde-desktop
  - 日本語を扱う**KDEデスクトップ環境**の推奨を規定
    - 特に日本語入力の推奨設定なし
    - task-japanese-desktopを継承するため
      ibus-mozcもしくはibus-anthy

      task-japanese-desktopでuim-mozcが除外されたので、
      既定がuim-mozcからibus-mozcに変更 (Debian 14 forky以降)

Xfce環境だったら？

- task-xfce-desktop
  - task-japanese-xfce-desktopは存在しない
  - task-japanese-desktopを継承するためibus-mozcもしくはibus-anthy
    - task-japanese-desktopでuim-mozcが除外されたので、
      既定がuim-mozcからibus-mozcに変更 (Debian 14 forky以降)

Debianにおける日本語入力関連パッケージ(まとめ)

- Debianにおける日本語入力(変換エンジン)はMozcを前提というのは変わらず
  - フレームワークはIBusがデファクトスタンダードに (非GNOME系が影響を受ける)
  - インプットメソッドのフレームワークは好みで選択できるのは変わらず
  - 変換エンジンはAnthyとかSKKとか選択肢はあるけど今回は割愛

本日の内容

- Debianにおける日本語入力関連パッケージ
- **UpstreamであるMozcの動向**
- DebianにおけるMozcのメンテナンス状況
- DebianのMozcを更新する試み
  - 更新の壁/どこまで視野にいれるか
  - Bazel対応の7つのポイント
  - 現在の状況と残作業

UpstreamであるMozcの動向

- 2010年: Mozcの公開
- 2023年10月以降のMozcに対する主な変更
  - https://zenn.dev/komatsuh/articles/komatsuh_mozc_updates_from_2023_10
    - ✅PRの受け入れ範囲の制限の撤廃

UpstreamであるMozcの動向(2)

- 2024 年 10 月から 2025 年 10 月までの Mozc に対する主な変更
  - https://zenn.dev/komatsuh/articles/komatsuh_mozc_updates_from_2024_10
    - ✅ビルドシステムがGYPからBazelに変更
    - ✅候補ウィンドウはGTK+からQtへ変更
    - ✅辞書・記号、絵文字データの更新
    - ✅変換内容に関する報告をGoogleフォームに移行

      Typing issue for Mozc https://forms.gle/oXgwbqzvSyVHNBSc7

本日の内容

- Debianにおける日本語入力関連パッケージ
- UpstreamであるMozcの動向
- **DebianにおけるMozcのメンテナンス状況**
- DebianのMozcを更新する試み
  - 更新の壁/どこまで視野にいれるか
  - Bazel対応の7つのポイント
  - 現在の状況と残作業

DebianにおけるMozcのメンテナンス状況

- https://tracker.debian.org/pkg/mozc
  - メンテナンスしてくれている”誰か”によるベストエフォート
  - 積極的にupstreamに追従しているわけではない
    - 2.28.4715.102 (2022/05/29 -)
    - 2.29.5160.102 (2024/12/20 -)

DebianにおけるMozcのメンテナンス状況(2)

- 2024/12/20
  - ついにMozc 2.28.4715.102から**2.29.5160.102**へ更新 🎉
- 2025年
  - FTBFSでMozcがtestingから消える
- 2026年
  - Ubuntu 26.04 (Resolute) からMozcが消える・消えないでごく一部で話題に

MozcがunstableでFTBFSになると困ること

- testingから削除 = Debian 14 (forky)でMozc利用不可
- 独自ビルドに支障がでる
  - 例: src/data/dictionary_oss/ を差し替えて語彙を増やす
    - https://github.com/utuhiro78/merge-ut-dictionaries
    - https://github.com/reasonset/mozcdict-ext.git

Mozcのリリースとの乖離
Debian採用バージョンより新しいもの

- Mozc (Upstream): **Debianでは最新リリースからは約3年遅れ**
  - 2.29系: **2.29.5160.102**, 2.29.5268.102, 2.29.5374.102, 2.29.5544.102
  - 2.30系: 2.30.5544.102
  - 2.31系: 2.31.5712.102, 2.31.5851.102
  - 2.32系: 2.32.5994.102
  - 3.33系: 3.33.6089, 3.33.6133
  - 3.34系: 3.34.6239

他のディストリビューションの状況

- openSUSE
  - 3.33.6133 (Bazelに移行済み)
- Fedora
  - 2.29.5111.102-20.fc45 (GTK renderer)

→ openSUSEが先行事例

本日の内容

- Debianにおける日本語入力関連パッケージ
- UpstreamであるMozcの動向
- DebianにおけるMozcのメンテナンス状況
- **DebianのMozcを更新する試み**
  - 更新の壁/どこまで視野にいれるか
  - Bazel対応の7つのポイント
  - 現在の状況と残作業

DebianにおけるMozcの更新の壁

- ビルドシステム変更の壁
- 依存ライブラリーの壁
- パッチ更新の壁

ビルドシステム変更の壁(影響:大)

- Mozcの開発元はビルドシステムをGYPからBazelに変更
  - 2.29.5544.102でGYPは非推奨になった
  - GYP対応は(それ以前からすでに)削除がはじまっている
    - 例: --use_gyp_for_ibus_buildは2.29.5160.102で削除済み
  - **GYPを維持しながらMozcを追従するのは非現実的**

ビルドシステム変更の壁(影響:大)

- Bazelは4.2.3までしかパッケージングされていなかった(2024年当時)
  - Bazel 7.xにMozcは当時対応していた
    - Bazelのパッケージングどうする？🤔
  - メーリングリスト(debian-bazel@lists.d.o)でも反応なし😭
    - https://lists.debian.org/debian-bazel/2024/07/msg00000.html
  - 2026-06-29: **Bazel 7.7.1がunstableへと投入された**

依存ライブラリーの壁(影響:中)

- Bazelがthird_partyのソースコード等を取得してビルドする
  - upstreamが想定するバージョンとDebianでパッケージ化されているバージョンは違うことが多い
  - **Debianのシステムライブラリーを使う方針と相性が悪い**

パッチ更新の壁(影響:中) (1/2)

- uim-mozcとfcitx-mozc,fcitx5-mozcはパッチにより
  src/unix配下にソースツリーを統合してビルドしている
  - fcitx-mozcのupstream (https://github.com/fcitx/mozc)
  - uim-mozcのupstream (https://github.com/e-kato/macuim)
  - GYP前提のパッチを捨てて、Bazelへ移行する必要

パッチ更新の壁(影響:中) (2/2)

- 既存のパッチ(19)の棚卸しが必要

```
0001-Support-uim.patch
0002-Support-fcitx.patch
0003-Change-compiler-from-clang-to-gcc.patch
0004-Add-usage_dict.txt.patch
0005-Enable-verbose-build.patch
0006-Update-gyp-using-absl.patch
0007-common.gypi-Use-command-v-instead-of-which.patch
0008-renderer-Convert-Gtk2-to-Gtk3.patch
0009-protobuf.gyp-Add-latomic-to-link_settings.patch
0010-Fix-the-compile-error-of-ParseCommandLineFlags-with.patch
0011-enable-deprecated-gtk-renderer.patch
0012-Enable-use_gyp_for_ibus_build-again.patch
0013-ibus-drop-needless-client_mock.patch
0014-protobuf-revert-internal-cleanup.patch
0015-uim-mozc-fix-ftbfs.patch
0016-Fix-FTBFS-with-absl-20240722.0.patch
0017-Fix-FTBFS-CPP20-FCITX5-RISCV.patch
0018-Fix-FTBFS-with-absl-20260107.0.patch
fcitx5-backport/e13338a41329f58e724801771f0cecf7338e24fb.patch
```

どこまで視野に入れて更新していくか？

- ビルドシステムはどのバージョンを前提にするか?
- GUIをどうするか?
- Mozcのベースバージョンをどれにするか?

どこまで視野に入れて更新していくか？(ビルドシステム)

- 時間的な制約
  - Debian 14 (forky)のフリーズが来年からはじまるはず
  - Bazel 8.xや9.xがforkyへ投入されるかは未知数(私見)
  - このままGYPベースで行くのは茨の道
    - ❌upstreamが放棄したものをメンテしていくコスト発生
    - ✅Bazel 7.7.1で対応していく

どこまで視野に入れて更新していくか？(GUI)

- 依存上の制約
  - Ubuntu 26.04対応でGTK3へ移植されたが、UpstreamはすでにQt
    - ❌GTK3の候補ウィンドウを維持する？
    - ✅Upstreamの変更(Qt)に追従

どこまで視野に入れて更新していくか？(Mozc)

- 依存上の制約
  - Mozc 3.34.6239はBazel 9.0.2前提
    - ❌Bazel 8.6と互換性を保っているとはあるがDebian unstableは7.7.1
  - Mozc 3.33.6133はBazel 8.4.1前提
    - ✅Bazel 8特有の機能(macroとか)を使っていなければ可能性がある(→使ってなかった)

Mozc 2.29から3.33に更新されると(1)

- ATOK辞書のインポート不具合修正(3.33.6133)
- 履歴よりも辞書候補を限定的に優先する(3.33.6133)
- 読み修正のインライン記法「巣(そう)窟」に対応(3.33.6089)
- **Wayland対応の改善**(Super+Space入力切り替え・候補ウィンドウ位置ズレ) (2.31.5712.102)
- 「なう」で日時が入力可能(2.30.5544.102)

Mozc 2.29から3.33に更新されると(2)

- 「れいわ8ねん」を西暦に変換できる(2.30.5544.102)
- Gboardの辞書取り込み対応(2.30.5544.102)
- 誤変換報告フォームのフィードバックが反映
- **パッチを減らせることによるメンテナンスコスト削減**

3.33.6133 をターゲットに やること

- パッチ更新の壁
  - 既存の19パッチの棚卸し
  - Uim向けパッチの更新
    - API変更にともなうFTBFS対応
  - Fcitx5対応パッチの更新
- ビルドシステム・依存ライブラリーの壁
  - Bazel 7対応

Bazel対応について

- 前提: DebianにおけるBazelサポートはまだこれから
  - debhelperのサポート(dh --buildsystem bazelみたいな)はない
  - Bazeliskの利用がMozcのビルドに推奨されているが、Debianにはない
  - → Bazel対応のありがちなポイント**7つ**を紹介

Bazel対応の7つのポイント(1/2)

- ネットワーク・ビルド環境
  - Q1. ダウンロードをやめさせるには
  - Q2. ビルド時のcacheをどこに逃がすか
- WORKSPACEからの移行
  - Q3. WORKSPACEの利用は非推奨、どのように対応するか

Bazel対応の7つのポイント(2/2)

- 依存モジュールの差し替え
  - Q4. Debianでパッケージ化されたBazelモジュールを使わせる方法
  - Q5. システムのライブラリーを使わせる方法
  - Q6. MODULE.bazelの依存関係を一部無効化する方法
- パッチ由来のリソース
  - Q7. パッチで追加したリソースをBazelに認識させるには

Q1. ダウンロードをやめさせるには

- A. bazel実行時に--repository_disable_downloadを指定する
  - --nofetchというオプションもあるが、(ネットワークアクセス制限下で)再ビルドする用途。
    リポジトリルール(依存解決)を事前に実施済みでないといけないので、Debianパッケージのビルドには向かない。

Q2. ビルド時のcacheをどこに逃がすか

- A. debian/rulesでexport XDG_CACHE_HOME=を指定する
  - もしくはbazelで--output_user_rootを指定する
  - 設定しておかないと権限無いところに保存しようとして失敗する

Q3. WORKSPACEの利用は非推奨 どのように対応するか

- A. bazel実行時に--noenable_workspaceを指定する
  - WORKSPACE → BUILD.bazel, MODULE.bazel前提に変更する
  - WORKSPACEはBazel 8で既定で無効化されているのを見据えた変更

Q4. Debianでパッケージ化されたBazelモジュールを使わせる方法

- A. bazel実行時に--override_module=(モジュール名)=(モジュールのパス)を指定する
  - e.g. --override_module=rules_license=/usr/share/bazel/rules/license

Q5. システムのライブラリーを使わせる方法

- A. bazel実行時に--override_moduleを指定する
  - e.g. --override_module=zlib=../debian/third_party/zlib

```
> cat debian/third_party/zlib/BUILD.bazel
package(default_visibility = ["//visibility:public"])

cc_library(
    name = "zlib",
    linkopts = ["-lz"],
)
> cat debian/third_party/zlib/MODULE.bazel
module(
    name = "zlib",
    version = "system",
)
```

Q6. MozcのMODULE.bazelに含まれる依存関係を一部無効化する方法

- A. 使っていない or パッケージ化されていないモジュールは --override_moduleで何もしないBUILD.bazelとMODULE.bazelを参照させる
  - e.g. --override_module=apple_support=../debian/third_party/apple_support

```
../debian/third_party/apple_support/BUILD.bazel は空のファイル
> cat ../debian/third_party/apple_support/MODULE.bazel
module(
    name = "apple_support",
    version = "system",
)
```

Q7. パッチで追加したリソースを Bazelに認識させるには

- A. リポジトリという単位でまとめる
  - BUILD.bazelとMODULE.bazelを配置したディレクトリを用意して--override_repositoryでその場所を指定する
  - e.g. usage_dict.txtとかアイコンのリソースとか
    - --override_repository=_main~_repo_rules~ja_usage_dict=third_party/ja_usage_dict

Bazel対応の7つのポイント(まとめ)

Q1. ダウンロードをやめさせるには
Q2. ビルド時のcacheをどこに逃がすか
Q3. WORKSPACEの利用は非推奨、どのように対応するか
Q4. Debianでパッケージ化されたBazelモジュールを使わせる方法
Q5. システムのライブラリーを使わせる方法
Q6. MODULE.bazelの依存関係を一部無効化する方法
Q7. パッチで追加したリソースをBazelに認識させるには

→だいたいbazelのオプション指定とレシピを用意すれば解決できる

現在の状況と残作業

- 現在の状況は?

途中経過報告(1/3)

- パッチの棚卸しと更新済み (19パッチ→10パッチ)

```
0001-Support-uim.patch
0002-Support-fcitx5.patch
0003-fix-mozc-visibility.patch
0004-fix-mozc-ftbfs.patch
0005-fix-uim-ftbfs.patch
0006-src-add-missing-pkg-config-fcitx5-and-uim.patch
0007-Add-usage_dict.txt.patch
0008-Add-material-icons.patch
0009-uim-add-BUILD.bazel.patch
0010-Add-zip-code-data.patch
```

途中経過報告(2/3)

Tracker – Changelog – Bugs – packages.d.o – Source

Package(s): mozc Suite: experimental Go

Compact mode Co-maintainers

Architecture	Version	Status	For	Buildd
all	3.33.6133+ds1-0.1~exp2	Installed	1d 13h 50m	x86-csail-02
amd64	3.33.6133+ds1-0.1~exp2	Installed	1d 14h 45m	x86-ubc-01
arm64	3.33.6133+ds1-0.1~exp2	Installed	1d 15h 15m	arm-conova-02
armhf is not present in the architecture list set by the maintainer				
i386 is not present in the architecture list set by the maintainer				
loong64	3.33.6133+ds1-0.1~exp2	Installed	1d 2h 30m	loongson-conova-03
ppc64el	3.33.6133+ds1-0.1~exp2	Installed	1d 15h 15m	ppc64el-conova-02
riscv64 ↓	3.33.6133+ds1-0.1~exp2	BD-Uninstallable	2h 20m	
s390x is not present in the architecture list set by the maintainer				
alpha is not present in the architecture list set by the maintainer				
hppa is not present in the architecture list set by the maintainer				
hurd-amd64 is not present in the architecture list set by the maintainer				
hurd-i386 is not present in the architecture list set by the maintainer				
m68k is not present in the architecture list set by the maintainer				
powerpc is not present in the architecture list set by the maintainer				
ppc64 is not present in the architecture list set by the maintainer				
sh4 is not present in the architecture list set by the maintainer				
sparc64 is not present in the architecture list set by the maintainer				
x32 is not present in the architecture list set by the maintainer				

- **experimental**に**Mozc 3.33.6133+ds1-0.1~exp2**をアップロード

debian

途中経過報告(3/3)

- 簡単な動作確認済み(unstable)
  - fcitx5-mozc on Budgie desktop(wayland)
  - ibus-mozc on GNOME (wayland)
  - uim-mozc on Xfce (❌辞書ツールとかはクラッシュします)
- ❌ 32bit環境(i386, armhf)はサポート終了(Bazelの制約)
  - ibus-anthyもanthy-unicodeに移行する話がでているが32bit環境でビルドできたとの報告あり

残作業(1/2)
技術的な課題

- uim-mozcの動作不良の解消
  - 辞書ツール等がクラッシュする(既知)
  - → そもそも維持するかの判断が先
- パッケージング内容の整理
  - --override_moduleまわり含めdebian/の見直し

残作業(2/2)
リリースまでの道のり

- より整理したバージョンをexperimentalへ投入
- experimentalからunstableへ
- Debian 14 (forky)のフリーズに間に合わせる
  - フリーズは来年からはじまるはず

DEMO

手伝ってほしい/確認したいこと

- 依頼: experimentalからインストールしてフィードバック
  - sudo apt install -t experimental ibus-mozc fcitx5-mozc
  - フィードバック先: Mozc 更新のwishlistバグ
    - https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=1085173
  - uimはまともに動作しないのは既知なのでその点は不要
- 確認: uim-mozcを使っている人は？🙋
  - そのうち、廃止されると困る人は？🙋
  - task-japanese-desktopからはずされたので必要性は低下

さいごに

- 👍 Bazel 7.7.1がunstableで利用可能になった
- 👍 Mozc **3.33.6133**のPoCをexperimentalに投入済み
  - 👎 将来的にMozcの32bit環境(i386, armhf)サポートは終了します
- 🤔 uim-mozcはまだ必要ですか？
- 次のSTEP
  - **より整理したバージョンを今後experimentalに投入予定**
  - forkyのフリーズまでにunstableへ