

Wineの進化と改良

Ubuntu 26.04 LTS リリース記念 オフラインミーティング 26.06

2026/06/20

さがわ (X @sagawa_aki)

自己紹介

- さがわ @sagawa_aki
 - Wineのコントリビューター (2009年～)
 - 年数件のパッチを投稿 (近年はペースダウン)
 - 2018年6月のオフラインミーティングでも登壇 (8年ぶり)

本日のトピック

- Wineの紹介
- Wineの動かし方
- Wineの進化
- Wineの改良

Wineの紹介

Wineとは

- POSIX対応環境でWindowsアプリケーションを動かす互換レイヤを提供するソフト
 - 動作OS: Linux (Android含む?)、macOS、FreeBSD、NetBSD
 - 動作プロセッサ: x86_64、i386、AArch32、AArch64
 - ライセンス: GNU LGPL 2.1
 - <https://www.winehq.org/>
- 商用製品CrossOverをCodeWeaversが開発
- 1993年から開発が続いている歴史あるオープンソースプロジェクト

UbuntuとWine

- Ubuntu (Universeレポジトリ) ではWineのバイナリパッケージを提供
- 25.10以降は新Wow64モードにより、32ビットライブラリが不要に

Ubuntuリリース	パッケージバージョン	備考
24.04 LTS	9.0~repack-4build3	
25.10	10.0~repack-6ubuntu1	新Wow64モードでビルド
26.04 LTS	10.0~repack-12ubuntu1	//

※最新版はWine 11.0 (安定版; 毎年1月提供)、および 11.11 (開発版; 隔週提供)

Wineのスゴイところ

- 以下を独自実装で実現 (全部OSS)
 - Windowsプログラムをメモリ上に読み込んで実行する仕組み
 - Windowsプログラムが利用する各種API (DirectX、DirectShow、XInput、VBScript、Windowsインストーラー、WMI、MCI、etc.)
 - Windowsプログラムが呼び出す実行ファイル (notepad.exe、rundll32.exe、regsvr32.exeなど)
 - フォント (Tahoma、Fixedsysなど)

Direct3D の実現

- DirectX/Direct3D (～11) をOpenGLに変換
- DirectX 11以降はVulkanに変換するDirect3Dドライバを利用
 - DXVK
 - DirectX 8～11をVulkanに変換
 - Philip "doitsujin" Rebohleさんが開発、別途インストール (Ubuntuパッケージあり)
 - VKD3D
 - DirectX 12をVulkanに変換
 - Wineと同じくCodeWeaversが主体で開発、Wineにも同梱

注: ここでの「ドライバ」は「動かす奴」ぐらいの意。デバイスドライバとは異なる

Protonについて

- 2018年8月に登場した互換レイヤ
- Valve (Steam提供元) とCodeWeaversが共同開発したゲーム特化型のWine
 - DirectX 8～11互換機能としてDXVKを利用
 - DirectX 12互換機能はVKD3D-Proton (VKD3Dがベース) を利用
 - dxvk-nvapi、メトリクス互換フォント、音声合成 (piper)、音声認識 (kaldi)
 - Steam API、その他ゲーム向けハック
- Steam Deckにも採用

動作アプリケーションの傾向

- だいたい動く
 - 吉里吉里 2、吉里吉里 Z で作られた作品
 - RPG ツクール、RPG Maker、WOLF RPGエディターで作られた作品
 - 秀丸、サクラエディタ、Mery
 - TMIDI Player (MIDI再生)、mp3tag (音楽ファイルタグ編集)、AviUtl (動画編集)
- 動かない
 - AviUtl2、ゆっくりMovieMaker4 (Wine未実装の機能を利用しているため)
 - カーネルドライバを使うソフト (ハードウェア制御、Android仮想マシン、ウイルス対策)

Wineの動かし方

Wineのインストール (Ubuntu版)

- wineパッケージを導入
- 使いたい機能に応じた共有ライブラリのパッケージをインストール
 - sane (ドキュメントスキャナ)
 - pcsc-lite (ICカード)

Wineのインストール(最新版)

- WineHQにて独自ビルドをリリースごとに提供
 - <https://gitlab.winehq.org/wine/wine/-/wikis/Debian-Ubuntu>
 - Ubuntu版Wineパッケージと共存不可
 - Wine 11.0(最新安定版)のUbuntu 26.04向けパッケージはない
 - 機能に応じた共有ライブラリのパッケージが必要になる点は同じ
- Gitやtar ballを利用してソースからのコンパイル
 - configure時に`--enable-archs=i386,x86_64`を加えることで新WoW64モード対応の構成になる

アプリケーション起動方法

- 「wine 実行ファイル名 引数」のようにして起動する
 - 初回起動時は初期設定が行われ、~/ .wine配下にファイルがコピーされる
 - Wine Mono、Wine Geckoの各パッケージを導入していない場合は、ダウンロードとインストールを促されるのでインストールする
- アプリケーションを終了するとWineも終了する
 - どうしても終了しないときは「wineserver -k」でkill

アプリケーションの導入方法

- アプリケーションの導入はWindowsと同じ
 - インストーラーの実行ファイル (*.exe)
 - `wine setup.exe`
 - Windowsインストーラー (*.msi)
 - `wine start setup.msi`
 - MSIXパッケージ (*.msix)
 - 未対応
 - ZIPファイル
 - Ubuntu、Wineのいずれかでファイルを展開
 - 展開後に実行ファイルを指定してWineで実行

文字化け対策

- 日本語フォント (MS ゴシック、メイリオなど) がいないため他のフォントを代替指定する
 - 自分でレジストリの `HKCU\Software\Wine\Fonts\Replacements` を設定する
または、
 - `winetricks fakejapanese` などを利用する
- 文字幅 (正確にはメトリクス) が異なるので、ダイアログボックスのレイアウトが崩れたり、テキストボックスに文字が入らないことがある
 - メトリクスを似せた「梅フォント」を代替指定するとトラブルが少ない

便利な環境変数

- WINEPREFIX
 - ~/.wine相当のディレクトリを指定
 - これを定義・変更すると複数環境を切り替えられる
- WINEARCH
 - 新WoW64モードでは廃止 (旧Wow64モードで32ビット専用環境の構築に利用)
- WINEDEBUG
 - デバッグ用チャンネルを指定
 - 0128:err:pidl:_ILSimpleGetText -- no text

Wineの進化

Wineの進化 - 5選

- ここ最近のWineにどのような進化があったか(ピックアップ)
 - PE化
 - 新WoW64モード(①)
 - ARM対応(ARM64EC含む)
 - NTSync(②)
 - Syscall User Dispatch
 - 翻訳の長音がWindows 7風に
 - HLSLコンパイラ(③)
 - DirectMusic実装(④)
 - ソートテーブルの置き換え(⑤)
 - MonoのANSI CP対応
 - Waylandドライバの提供
 - Bluetooth
 - リパースポイント対応
 - バッチファイル/VBScript強化

①新WoW64モード

Wine 9.0～11.0 (2024～26)

- 課題

- 32ビットUNIXライブラリなしで32ビットプログラムを動かしたい

- 解決内容

- PE化によりUNIX側のコード処理を分離(入口を整理)
 - UNIX側のコードは64ビットのみを実現
- OSとCPUの助けを借りて64ビットコードと32ビットコードをプロセスのメモリ内で共存させて実行(set_thread_area(2)でLocal Descriptor Tableを変更)
- ビット幅が異なる値は中間層で相互変換、ポインタは頑張る

注: PEとはPortable Executableの略。Windowsで実行ファイルに使われている形式

②NTSync

Wine 11.0 (2026)

- 課題
 - wineserverにあるWindows NTの同期機構の効率が悪い
- 解決内容
 - Linuxカーネルドライバntsyncを実装
 - カーネルとの直接対話でwineserverでの調停が不要に
 - Ubuntu 26.04ではカーネルモジュールが標準提供され広く使えるように

③ HLSLコンパイラ

Wine 10.0~11.0 (2025~26)

- 課題
 - シェーダー言語 (HLSL) のコンパイラがなかった
- 解決内容
 - VKD3DにHLSLコンパイラを実装
 - Shader Model 1, 2, 3 (Direct X 9世代) もOK
 - MS純正のd3dcompiler_*.dllの導入が不要に
- 以前の発表で挙げた「悩ましいバグ」その1

④ DirectMusic

Wine 9.0～11.0 (2024～26)

- 課題
 - DirectMusic (ソフトウェアMIDI) が動かなかった
- 解決内容
 - FluidSynth (ソフトウェアMIDI) のWindows版を丸抱えして実装
 - サウンドフォント(*.sf2)は別途導入する (Ubuntuパッケージあります)
 - MS純正のDirectMusicの導入が不要に
- 注: マルチメディアAPIは引き続きALSAで見えるMIDIデバイスが必要

⑤ソートテーブルの置き換え

Wine 8.0 (2023)

- 課題
 - 文字列の並び替え順序がWindowsと違う(二分探索などで問題に…)
- 解決内容
 - Windows Sortkeyデータベース (MS提供) を組み込み
 - 並び替えのアルゴリズムも実装
 - Windows 11 (「令和」対応) はWine 11.10以降
- 以前の発表で挙げた「悩ましいバグ」その2

Wineの改良

Wineは完璧ではない

- Windowsと同じように動かないならばWineのバグ(原則)
 - とはいえ無理なものは無理
 - アプリケーションの作りが悪い場合(メモリの二重開放など)
 - winetricksでMSの純正ファイルを使う(ソフトウェアライセンスに注意!)
 - とりあえずwinetricksは悪手
 - Wineを少し修正することで動くこともある
 - こっちを頑張る(=バグ修正)
- 以降でバグ修正の過程を7ステップで概説

バグ修正の流れ(1/7)

- STEP 1: 対象となるバグを見つける
 - 自分で見つける
 - アプリケーションを動かして気になる挙動がないか観察
 - Xの投稿を目にする
 - Yahoo!リアルタイム検索でエゴサーチ
 - バグレポート (Bugzilla) を見る

バグ修正の流れ(2/7)

- STEP 2: 手元でバグを再現する
 - 入手困難なソフトはこれができないので大変(やりたくない)

バグ修正の流れ(3/7)

- STEP 3: バグの原因を探る
 - 未実装機能は比較的特定しやすい
 - 環境変数WINEDEBUGを利用してAPI呼び出しを追跡
 - Wineソースコードにデバッグ用メッセージを追加
 - 低レベルデバッガ (x64dbgなど) でステップ実行
 - 解析が禁止されていなければ逆アセンブラ (Ghidra) を使うことも
 - MSの実装は参照禁止

バグ修正の流れ(4/7)

- STEP 4: 修正方法を考える
 - APIリファレンスから挙動を考える
 - 適合性テストや小さなテストプログラムで動作を確認する
 - 周辺のソースコードから実装方法を検討する

バグ修正の流れ(5/7)

- STEP 5: 修正を実装してテストコードが動くようにする
 - 実装を書く (C言語)
 - LLM (生成AI) は利用不可
 - Wineの開発ガイドラインで禁止
 - バグレポートを作成する

バグ修正の流れ(6/7)

- STEP 6: Wine GitLabに投稿する
 - 自分のリポジトリにpushしてマージリクエスト(プルリクエスト相当)を作成
 - メインテナーのコードレビューを待つ
 - 指摘箇所を修正する

バグ修正の流れ(7/7)

- STEP 7: 修正が開発レポジトリにマージされる
 - 問題なければ次期Wineリリースに含まれて提供される

直近の貢献

- MPEG-1動画再生処理 (DirectShow) を改善
 - 某ゲームの再生処理を改善 (Bug 56485、2024年5月)
 - ループする動画が動くようにシーク処理を改善 (Bug 55933、2026年3月)
 - 再生終了の検出に使うAPI (IMediaEventEx::WaitForCompletion) が2回目以降動作しないケースを修正 (2026年4月)
 - ループ後先頭に戻ったときのカクつきをなくす (WIP)

Wineの今後

- 公式ロードマップがないので勝手予想
- 安定版のポイントリリース (Wine 11.0.x) はない
 - 2025年3月～メンテナー不在
- 2027年1月 Wine 12.0リリース
 - 例年のスケジュール
 - 2026年12月:コードフリーズ
 - 2027年1月:リリース
- 乞うご期待!

まとめ

- 今日のおさらい

- Wineの紹介
- Wineの動かし方
- Wineの進化
- Wineの改良

- ご清聴

ありがとうございました

-  @sagawa_aki