

# 異なる複数のOS向けバイナリが実行可能なOS Scarlet

鬼頭 太郎

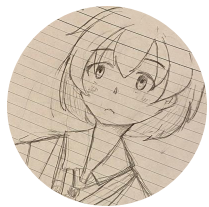
サイボウズ・ラボユース成果発表会2026

2026/03/26

# 自己紹介

鬼頭 太郎 (きとう たろう)

- 立命館大学 M1 情報理工学研究科, M研究室
  - RISC-Vのハイパーバイザに関する研究
- 趣味
  - 情報系全般
    - 自作OS, オーディオルーター&ミキサー, 色々なツール, etc...
    - 自宅鯖, ヤフオク漁り, おうちクラウド(k8s), etc...
  - 音楽
    - フルート, ベース, シンセ



X: @petitstb

GitHub: petitstrawberry



A kernel in Rust designed to provide a universal, multi-ABI container runtime.と銘打ってますが  
その実態は思いついたことを詰め込んだキメラ

- 複数OS向けバイナリの透過的な実行

- ユーザやプログラムは何も意識せずに  
execやshellを使うだけで、  
色々なOS向けのバイナリが動く
  - 異なるOSバイナリ間のIPCなども可能

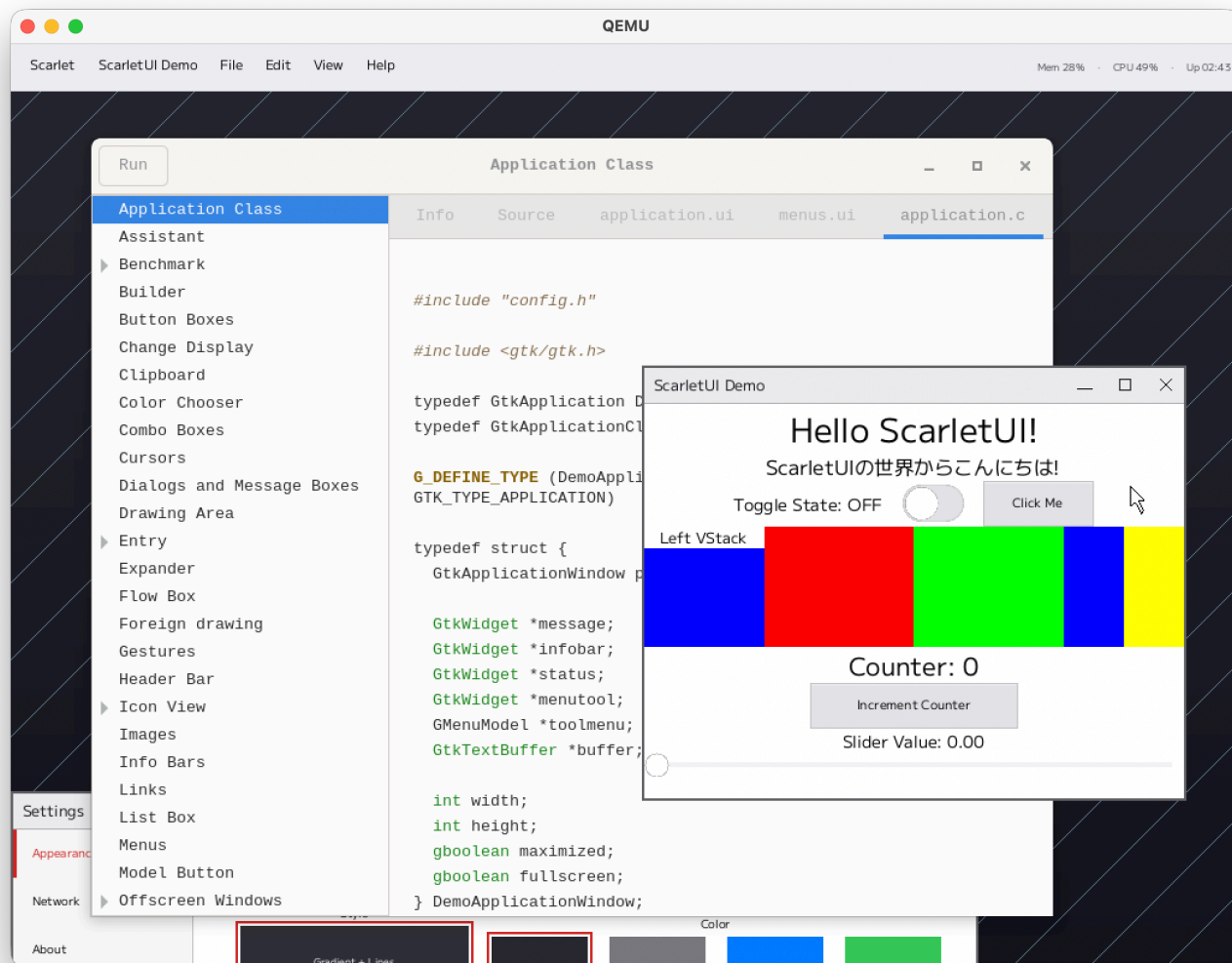
- 分離と共有の両立

- OSバイナリごとに前提とする環境は  
さまざま
- fsなどのリソースの分離と共有を上手く  
やることで、実現性と利便性の両立
- バイナリごとに異なるコンテナで動きつつ、  
homeディレクトリなどをbind mountしてる  
イメージ

- Rust製

- 国内・個人だとそこそこ大規模な方かも

[1] Scarlet: <https://github.com/petitstrawberry/Scarlet>



# Kernel Architecture

カーネルは異なるOS向けバイナリを実行するための共通の基盤を提供する

- Kernel Core

- カーネルの基本的な機能を提供
- Kernel Objectとして抽象化
  - FileObject, StreamOps, PipeObject, MemoryMappingObject, etc...  
トレイトを実装したオブジェクト群

- ABI Module

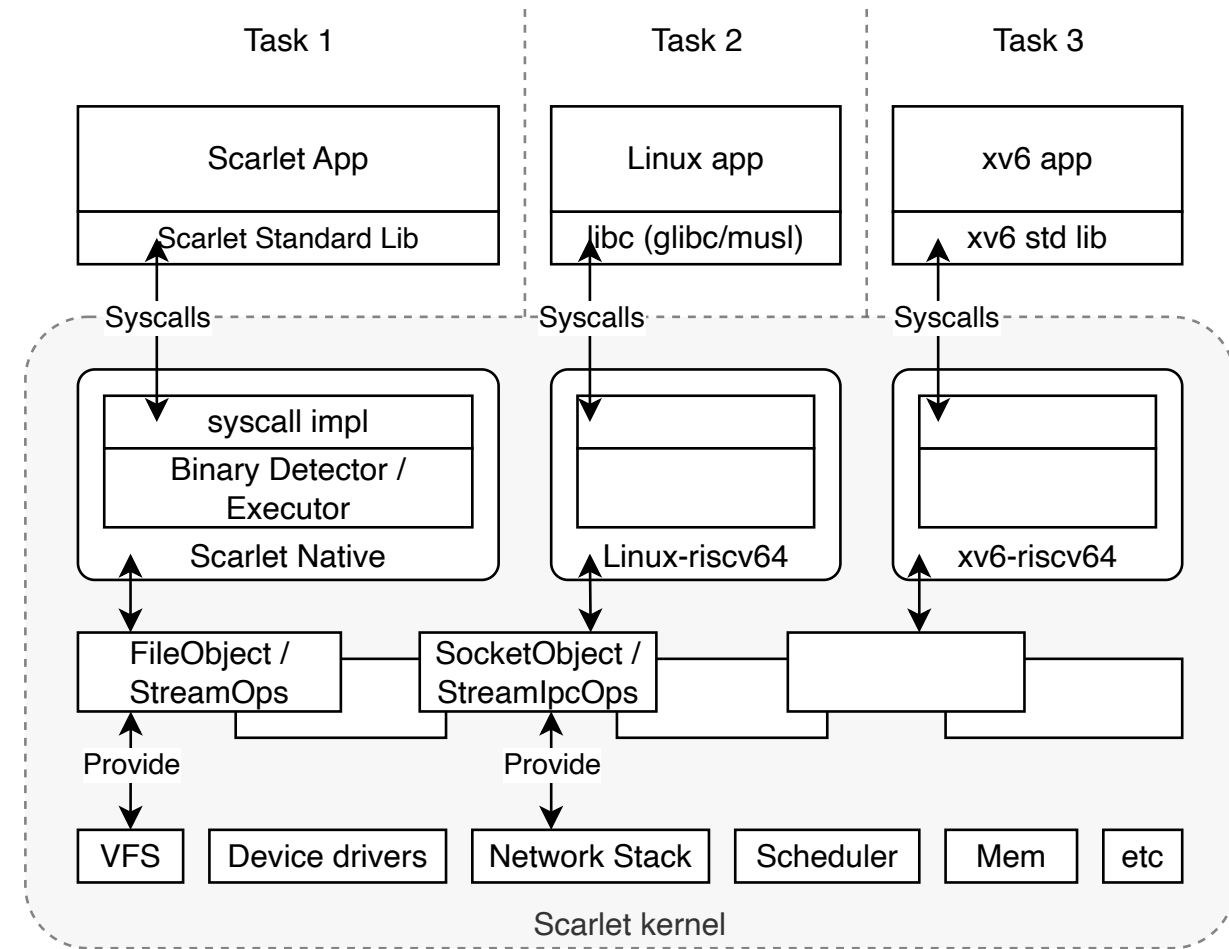
- OS/ABIごとに異なる環境を提供
- バイナリの種別判定, 実行環境構築
- Syscall, 特殊なファイルの実装など
- Kernel Objectを利用して, 実装

User Application

ABI Module

Kernel Object

Kernel Core



簡単にいうと, Kernel CoreがOSに依存しない共通の基盤を提供して, ABI ModuleにOS固有の処理や環境構築を隠蔽するイメージ

実行するバイナリごとに異なるrootfsや標準ライブラリを使用する

- Scarlet Native
  - rootfs
    - Scarlet独自のバイナリ群や設定ファイルなどを配置
  - 標準ライブラリ
    - rust製の標準ライブラリ, LinuxのRustの標準ライブラリと同じようなAPIを提供
- Linux
  - rootfs
    - Linux向けに用意したrootfsを**そのまま**使用可能
    - デフォルトではbuildrootで生成したものを利用
  - 標準ライブラリ
    - musl推奨だが, glibcも使える? /使えるようになるはず (もちろんdynamic linking)
- xv6
  - rootfs/標準ライブラリ: xv6向けに用意したものをそのまま使用可能

# Scarlet Window System/Server (SWS) & Wayland Bridge

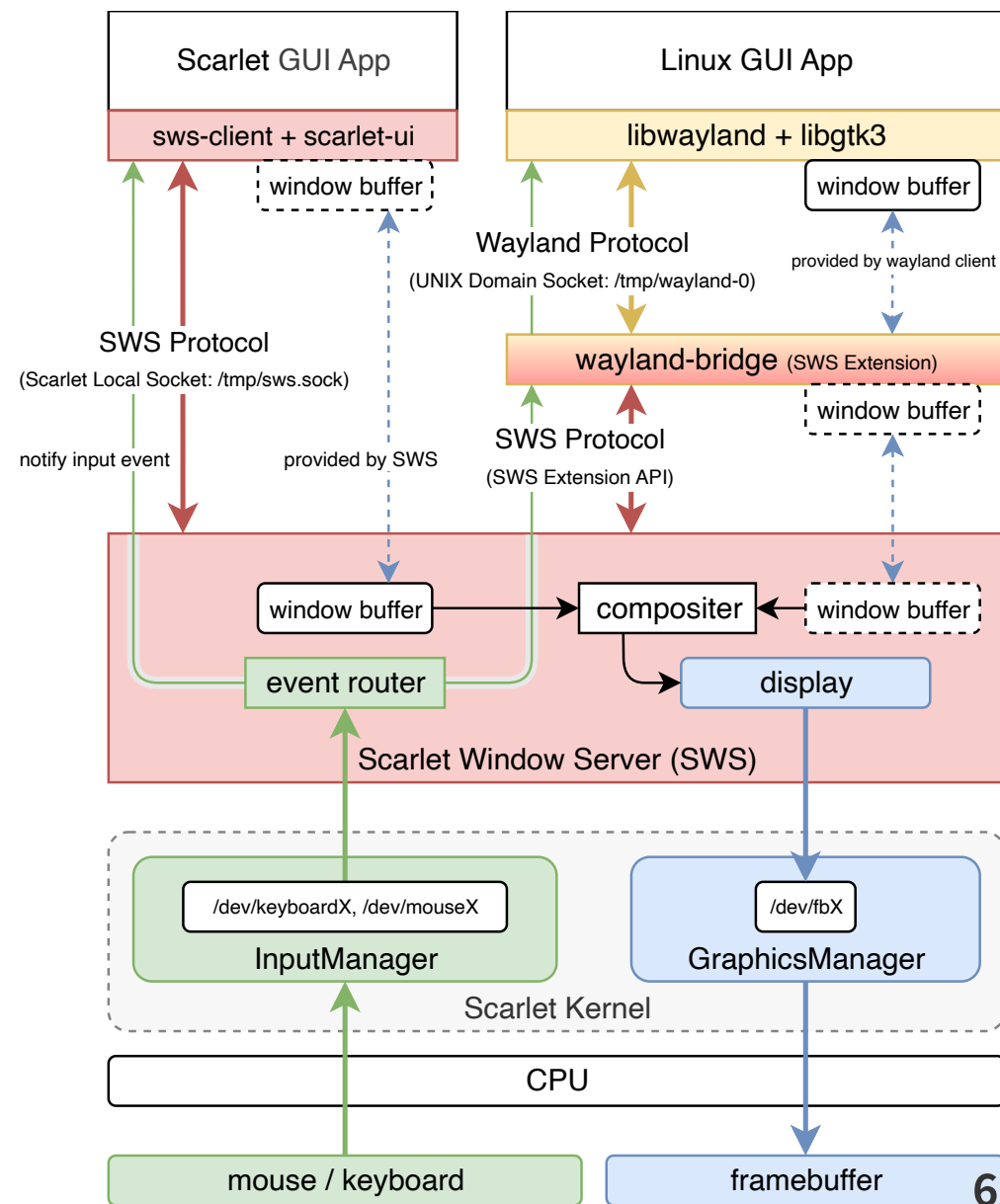
もちろんGUIも使える

## Scarlet Window System/Server (SWS)

- Scarlet独自のウィンドウシステム/サーバ
- Scarlet Nativeで動作
- SWS Client
  - SWSと通信してウィンドウを管理するクライアント

## Wayland Bridge

- SWSとWaylandクライアントの橋渡しをする
- WaylandクライアントがSWS上で動作できる
  - つまり... Linux向けのGUIアプリも動く
  - あとでデモ



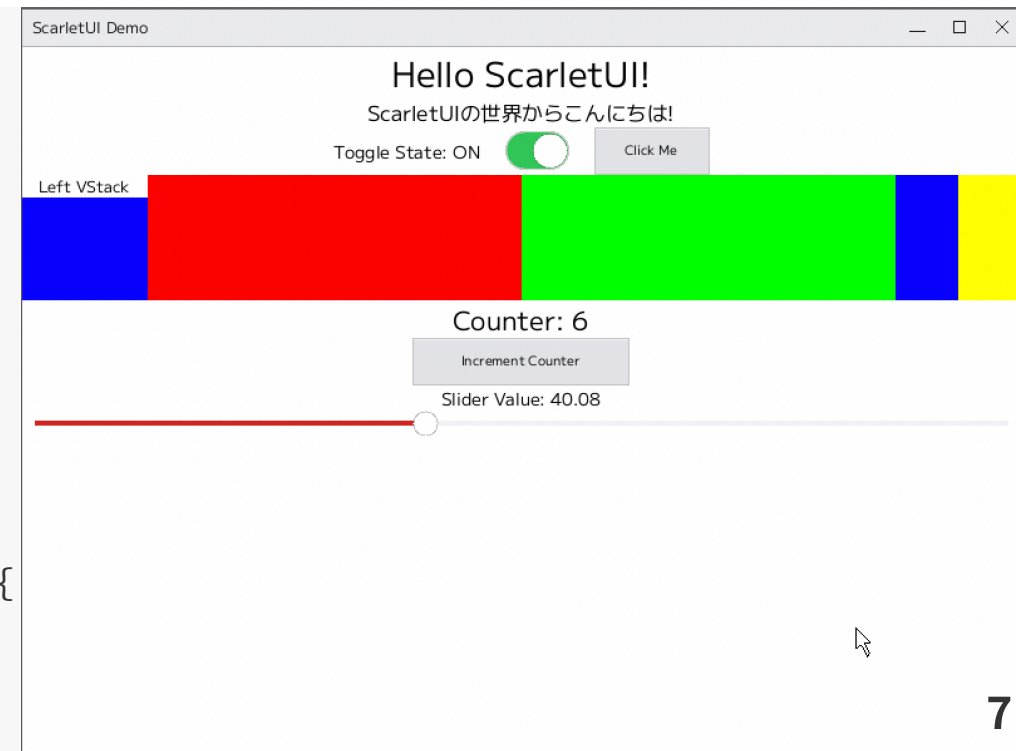
# Scarlet UI (おまけ)

ついでに宣言的UIフレームワークも作った (そこそこ形になったので紹介)

## Scarlet UI

- Swift UI / Flutterライクな宣言的UIフレームワーク
- レイアウトはもちろん, 変数の監視やイベントの監視, 自動差分更新などもやる

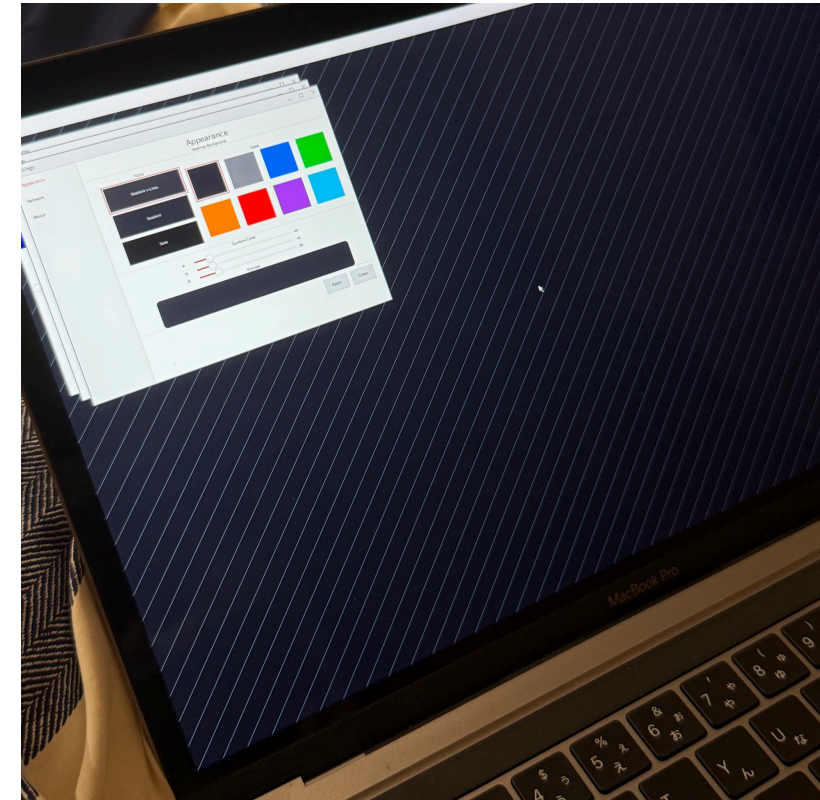
```
impl Application for DemoApp {  
  fn body(&self) -> impl View {  
    Window::new("ScarletUI Demo",  
      vstack! {  
        Text::new("Hello ScarletUI!")  
          .font_size(40.0),  
        Text::new("ScarletUIの世界からこんにちは!")  
          .font_size(24.0),  
        hstack! {  
          Text::new(format!("Toggle State: {}",  
            if self.toggle_state.get() { "ON" } else {  
              .font_size(20.0),  
          Spacer::new(),  
          Toggle::new(self.toggle_state.clone()),  
        }  
      )  
    }  
  }  
}
```



- **Scarlet Build System / Fully Modular Kernel (WIP)**
  - ABI moduleやDriver, そのほかScarletの膨大な機能をRustのcrateで管理
  - 必要な機能だけ有効にしてビルドする仕組みを膨大なfeature管理ではなく, いい感じにのtomlで管理できるようにするビルドシステム
  - 例えば, Linuxのアプリを動かす軽量ランタイムとして使いたい (Unikernel的な)場合
    - Linux ABI moduleのみ有効で, そのバイナリで使わないドライバは除外など
    - Unikernelで良いじゃんという話だが, ここは突っ込まないで
- **Scarlet Hypervisor (SHV)**
  - 仮想化機能を提供するType-2ハイパーバイザ, Scarlet on Scarletまでは確認した
  - KVMのような感じで, デバイスエミュレーションはユーザランドで
  - ただし... これもあくまでABI moduleにkernel Coreが提供する機能の一つであり, 直接Syscallは提供しない
    - Linux ABI moduleが**KVMの再現**に利用するイメージ
    - 近い将来, **ゲストOS on QEMU (KVM有効) on Linux ABI (/dev/kvmをエミュレーション) on Scarlet**が可能になる
- そのほか書ききれないもの
  - カーネル: TCP/IP, cgroupやname space的なやつ, ユーザランドにバイナリ実行を委譲する機能, etc...
  - ユーザランド: Stemd (systemd的な), sbud (dbud的な), Scarlet Package Manager (SPM), Webサーバ, etc...

# 対応するアーキテクチャ / ボード / ABI module

- 対応するアーキテクチャ
  - riscv64 / aarch64 (Tier 1, 比較的安定)
  - x86\_64 (Tier 2, WIP)
    - これが実現次第, windows-x86\_64-abi moduleも開発予定 (ScarletでWindowsバイナリが動くようになる)
- 動作確認済みボード
  - 現在は主にLimine<sup>[2]</sup>を利用してUEFI環境でブートする形で開発中
  - QEMU Virt machine (riscv64, aarch64)
  - SiFive HiFive Premier P550 (riscv64) [WIP]
  - Apple M1 MacBook Pro (aarch64) [WIP]
    - 割込みコントローラ, UART, framebufferなどが利用可能
    - Asahi Linux<sup>[3]</sup>のリバースエンジニアリングの成果を活用しています
    - M1 Macは良いARM開発ボード (笑) なのでおすすめ
- ABI module (実行可能なOSバイナリ)
  - Linux (riscv64, aarch64), xv6 (riscv64), Scarlet (riscv64, aarch64)



[2] Limine: <https://codeberg.org/Limine/Limine>

[3] Asahi Linux: <https://asahilinux.org/>

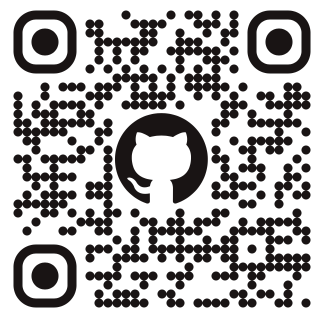
ここで, デモ

異なる複数のOS向けバイナリが実行可能なOS Scarletの紹介でした

- カーネルの互換だけでなく, ユーザランドも含めた高機能な互換環境を提供
- Linuxやxv6, Scarlet独自のバイナリなど, 異なるOS向けのバイナリが協調して動くOSを目指して動く (GUIアプリも)

## 発表など

- セキュリティキャンプ2025 LT会 「実は簡単！？自作OSでDOOMを動かそう！」
- セキュリティキャンプ2026 フォーラム展示
- 予定(Kernel/VM探検隊@関西 12回目 ????? 発表するかも ?????)



ScarletのGitHubリポジトリ

<https://github.com/petitstrawberry/Scarlet>

- 簡単に試せます / コントリビュートお待ちしております
- 複数アーキテクチャやボード対応が一人では厳しくなってきたので...ぜひ