

KeTCindyInstaller

KeTCindy 用パッケージマネージャの開発

秋本蒼空

木更津工業高等専門学校 情報工学科 3 年

2024 年 10 月 3 日

KeTCindyInstaller

2024-10-03

KeTCindyInstaller

KeTCindy 用パッケージマネージャの開発

秋本蒼空

木更津工業高等専門学校 情報工学科 3 年

2024 年 10 月 3 日

これから発表を始めます。よろしくお願いします。

はじめに

まずはじめに，自己紹介をさせていただきたいと思います．

秋本蒼空 (AKIMOTO Sora)

所属	木更津工業高等専門学校 情報工学科
部活動・同好会	プログラミング研究同好会 会長
好きな技術	Lua ^A T _E X, Vim, Rust, Gentoo Linux
最近興味のある分野	セキュリティ, インフラ・ネットワーク系
Website	https://nxvzbgbfben.pages.dev/

2024-10-03

KeTCindyInstaller
└─はじめに

└─はじめに

木更津高専 情報工学科 3 年の秋本蒼空と申します。
今年度からプログラミング研究同好会の会長を務めさせていただいております。
個人的に好きな技術については Lua^AT_EX, エディタの Vim, プログラミング言語の Rust, OS の Gentoo などがあります。
最近はセキュリティに興味があって CTF に出場することが多いです。インフラやネットワーク系の方にも興味があって、この夏休みはインフラ系の企業のインターンに参加しました。

秋本蒼空 (AKIMOTO Sora)

所属	木更津工業高等専門学校 情報工学科
部活動・同好会	プログラミング研究同好会 会長
好きな技術	Lua ^A T _E X, Vim, Rust, Gentoo Linux
最近興味のある分野	セキュリティ, インフラ・ネットワーク系
Website	https://nxvzbgbfben.pages.dev/

KeTCindy プロジェクトには 2023 年 2 月頃参加

- KeTCindy スタートブック
- KeTLMS の課題一斉送信スクリプト (開発中)
- **KeTCindyInstaller (開発中)**

KeTCindyInstaller

└─はじめに

└─はじめに

2024-10-03

KeTCindy プロジェクトには 2023 年の 2 月頃から参加させていただいております。

KeTCindy のインストール方法を解説した「KeTCindy スタートブック」であったり、

KeTLMS の課題を学生に一斉送信するスクリプトであったり、今回ご説明させていただく KeTCindyInstaller を開発しています。

はじめに
自己紹介

秋本蒼空 (AKIMOTO Sora)

所属	木更津工業高等専門学校 情報工学科
部活動・同好会	プログラミング研究同好会 会長
好きな技術	Lua ^A T _E X, Vim, Rust, Gentoo Linux
最近興味のある分野	セキュリティ, インフラ・ネットワーク系
Website	https://nxvzbgbfben.pages.dev/

KeTCindy プロジェクトには 2023 年 2 月頃参加

- KeTCindy スタートブック
- KeTLMS の課題一斉送信スクリプト (開発中)
- **KeTCindyInstaller (開発中)**

開発について

2024-10-03

KeTCindyInstaller
└─ 開発について

開発について

次に，開発についてご説明させていただきます．

KeTCindy は環境構築が大変
→ Windows・Mac 共通のインストーラがあると便利

KeTCindyInstaller
└─ 開発について

2024-10-03

└─ 開発について

KeTCindy は環境構築が大変であるという現状を踏まえ、Windows および macOS 共通のインストーラがあると便利だと思っています。

KeTCindy は環境構築が大変

→ Windows・Mac 共通のインストーラがあると便利

KeTCindyInstaller

(<https://github.com/NXVZBGBFBEN/KeTCindyInstaller>)

- KeTCindy と関連ソフトウェアをセットアップ
- クロスプラットフォームで動作 (Windows・Mac 対応)
- CLI アプリケーション→自動化が容易

KeTCindyInstaller

└─ 開発について

└─ 開発について

2024-10-03

そこで、KeTCindyInstaller の開発を開始しました。

KeTCindy と、R や Maxima などの関連ソフトウェアをセットアップできるアプリケーションを目指しており、クロスプラットフォームで動作することを強みにしたいと考えています。また、コマンドラインで動作するアプリケーションなので、シェルスクリプトからの呼び出しなど、自動化が容易であるという特徴も備えています。

開発について

KeTCindy は環境構築が大変
→ Windows・Mac 共通のインストーラがあると便利

KeTCindyInstaller
(<https://github.com/NXVZBGBFBEN/KeTCindyInstaller>)

- KeTCindy と関連ソフトウェアをセットアップ
- クロスプラットフォームで動作 (Windows・Mac 対応)
- CLI アプリケーション→自動化が容易

KeTCindy は環境構築が大変

→ Windows ・ Mac 共通のインストーラがあると便利

KeTCindyInstaller

(<https://github.com/NXVZBGBFBEN/KeTCindyInstaller>)

- KeTCindy と関連ソフトウェアをセットアップ
- クロスプラットフォームで動作 (Windows ・ Mac 対応)
- CLI アプリケーション→自動化が容易

現状インストール機能の実現を目指して開発中だが、将来的にはパッケージのアップデートやアンインストールにも対応させたい

KeTCindyInstaller

└─ 開発について

└─ 開発について

2024-10-03

現状ではまずパッケージのインストールができるように開発を進めていますが、
最終的にはアップデートやアンインストールといった、一種のパッケージマネージャとして成長させていきたいと考えています。

開発について

KeTCindy は環境構築が大変
→ Windows ・ Mac 共通のインストーラがあると便利

KeTCindyInstaller
(<https://github.com/NXVZBGBFBEN/KeTCindyInstaller>)

- KeTCindy と関連ソフトウェアをセットアップ
- クロスプラットフォームで動作 (Windows ・ Mac 対応)
- CLI アプリケーション→自動化が容易

現状インストール機能の実現を目指して開発中だが、将来的にはパッケージのアップデートやアンインストールにも対応させたい

開発言語には Rust を採用

- パフォーマンスに優れる
- クロスコンパイルが可能
- 開発者の負担が少ない
- バイナリ単体で動作

開発言語には Rust を採用しました．理由としては，一般にパフォーマンスがよいと言われていること，クロスコンパイルが可能であること，エラー処理が容易な文法によって開発者の負担が軽減されていること，プラットフォームネイティブなバイナリを生成できることが挙げられます．

開発言語には Rust を採用

- パフォーマンスに優れる
- クロスコンパイルが可能
- 開発者の負担が少ない
- バイナリ単体で動作

できる限り依存関係は少なくしたい

→ ライブラリを静的リンクしてコンパイル

→ Windows でありがちな「VC++ランタイムがない」を回避

KeTCindyInstaller

└─ 開発について

└─ 開発について

2024-10-03

また，KeTCindyInstaller 自体の依存関係をできるだけ少なくするため，本来は動的リンクされる Visual C++ Runtime 等のライブラリも静的リンクするようにしています．

これにより，Windows でありがちな「ランタイムが存在しないので動作しない」といった事態を避けることができます．

開発について
設計

開発言語には Rust を採用

- パフォーマンスに優れる
- クロスコンパイルが可能
- 開発者の負担が少ない
- バイナリ単体で動作

できる限り依存関係は少なくしたい

→ ライブラリを静的リンクしてコンパイル

→ Windows でありがちな「VC++ランタイムがない」を回避

開発言語には Rust を採用

- パフォーマンスに優れる
- クロスコンパイルが可能
- 開発者の負担が少ない
- バイナリ単体で動作

できる限り依存関係は少なくしたい

- ライブラリを静的リンクしてコンパイル
 - Windows でありがちな「VC++ランタイムがない」を回避

パッケージ情報は別ファイル（マニフェスト）で管理

- プログラム自体を更新しなくてもパッケージの更新に追従可

そして、KeTCindyInstaller で扱うパッケージは全てマニフェストファイルとして情報が分離されており、リポジトリの別ブランチで管理されています。これにより、パッケージのダウンロードリンクが変更された場合にアプリケーションを再度ビルドする必要がありません。

- パフォーマンスに優れる
- クロスコンパイルが可能
- 開発者の負担が少ない
- バイナリ単体で動作

ソースコードのバージョン管理には Git(GitHub) を使用

- lefthook を用いて pre-commit チェック
- GitHub Pages ・ GitHub Actions を用いてドキュメンテーションを自動化
- GitHub Actions を用いた自動ビルド ・ リリース準備

ソースコードのバージョン管理には Git を採用しています。

lefthook というツールを使ってコミット前に format と lint を実行させることができたり，GitHub Actions を用いて rustdoc を実行することでパッケージのドキュメントを自動で GitHub Pages 上にデプロイできたり，自動で各プラットフォーム向けにクロスビルドしてリリースの準備を簡略化したりできるシステムを構築しています。

- lefthook を用いて pre-commit チェック
- GitHub Pages ・ GitHub Actions を用いてドキュメンテーションを自動化
- GitHub Actions を用いた自動ビルド ・ リリース準備

ソースコードのバージョン管理には Git(GitHub) を使用

- lefthook を用いて pre-commit チェック
- GitHub Pages ・ GitHub Actions を用いてドキュメンテーションを自動化
- GitHub Actions を用いた自動ビルド ・ リリース準備

Rust のエコシステムを活用して自動化

- Rustdoc を用いたドキュメンテーション
- rust-analyzer と Clippy を用いた format ・ lint

これらのシステムの背後には Rust の豊かなエコシステムがあり, Rust を採用する理由のひとつとなっています.

Rustdoc を利用することで API ドキュメントを簡単に生成でき, rust-analyzer と Clippy を使うことでコード品質を保つことができます.

- lefthook を用いて pre-commit チェック
- GitHub Pages ・ GitHub Actions を用いてドキュメンテーションを自動化
- GitHub Actions を用いた自動ビルド ・ リリース準備

- Rustdoc を用いたドキュメンテーション
- rust-analyzer と Clippy を用いた format ・ lint

ソースコードのバージョン管理には Git(GitHub) を使用

- lefthook を用いて pre-commit チェック
- GitHub Pages ・ GitHub Actions を用いてドキュメンテーションを自動化
- GitHub Actions を用いた自動ビルド ・ リリース準備

Rust のエコシステムを活用して自動化

- Rustdoc を用いたドキュメンテーション
- rust-analyzer と Clippy を用いた format ・ lint

共同開発も視野に入れ環境を構築中

さらに今後は Pull Request 発行時の自動テストを含め、共同開発も視野に入れた開発環境づくりをしています。

- lefthook を用いて pre-commit チェック
- GitHub Pages ・ GitHub Actions を用いてドキュメンテーションを自動化
- GitHub Actions を用いた自動ビルド ・ リリース準備

- Rustdoc を用いたドキュメンテーション
- rust-analyzer と Clippy を用いた format ・ lint

現在、マニフェストのパスとパッケージのダウンロードまで完成
v1.0.0-alpha.1 として GitHub で公開中

KeTCindyInstaller
└─ 開発について

2024-10-03

└─ 開発について

現在は各パッケージのマニフェストファイルのパーサが完成し、
マニフェストに記載されたダウンロードリンクを用いてパッケージをダウン
ロードするところまで実装が完了しています。
GitHub 上で v1.0.0-alpha.1 として公開していますので、確認してみてください。

現在、マニフェストのパスとパッケージのダウンロードまで完成
v1.0.0-alpha.1 として GitHub で公開中

今後の課題

- マニュアル (README.md) の作成
- ライブラリを静的リンクするように変更 (GitHub Actions)
- パッケージのインストール
 - OS に強く依存する部分
- パッケージのアップデート
- パッケージのアンインストール
- ketcindysettings.cdy の処理内容サポート

開発について

開発について

2024-10-03

今後の課題として挙げられるのは、まず KeTCindyInstaller のマニュアルを作成することと、現状自動ビルド環境ではライブラリの静的リンクは設定されてないのでそれを設定することで、その後にパッケージのインストール部分を作成することを目指しています。パッケージのインストールについては処理が OS に強く依存しているので、少し実装が大変かなと思います。
その後、パッケージのアップデートやアンインストールに対応させ、最終的には ketcindysettings.cdy の処理も担っていきたいと思います。

さいごに

最後に、今後についてお話ししたいと思います。

KeTCindyInstaller は現状 Windows 上で開発されている
MacOS での動作確認を進めたい

KeTCindyInstaller
└─ さいごに

└─ さいごに

現状，KeTCindy は Windows 上で開発されているため，MacOS 上での動作確認ができていません．
今後は，実機環境での動作確認を進めていきたいと思
います．

KeTCindyInstaller は現状 Windows 上で開発されている
MacOS での動作確認を進めたい

KeTCindy は Cinderella と関連パッケージを用いて高度な処理ができる
より導入しやすい環境を作ること発展に貢献していきたい

2024-10-03

KeTCindyInstaller

└─さいごに

└─さいごに

KeTCindy は Cinderella とその関連パッケージを用いることで高度な処理を可能としています．より導入しやすい環境づくりを通して発展に貢献していきたいと思います．

さいごに
今後について

KeTCindyInstaller は現状 Windows 上で開発されている
MacOS での動作確認を進めたい

KeTCindy は Cinderella と関連パッケージを用いて高度な処理ができる
より導入しやすい環境を作ること発展に貢献していきたい

ありがとうございました

2024-10-03

KeTCindyInstaller
└─ さいごに

以上で発表を終わります．ありがとうございました．

ありがとうございました