

1. 担当 PM

竹迫 良範（株式会社リクルート データプロダクトユニット ユニット長）

2. クリエータ氏名

矢尾田貴大（Pianoforte 代表）

3. 委託金支払額

2,736,000 円

4. テーマ名

シェルスクリプトへのコンパイルを行う静的型付けスクリプト言語の開発

5. 関連 Web サイト

- Web サイト / ドキュメント : <https://cotowali.org>
- GitHub リポジトリ : <https://github.com/cotowali/cotowali>
- Konryu の GitHub リポジトリ : <https://github.com/cotowali/konryu>

6. テーマ概要

本プロジェクトでは、書きやすい構文を持ちながら、シェルスクリプトに変換することで高いポータビリティを実現する静的型付きスクリプト言語 Cotowali を開発する。Cotowali を利用することで、ポータビリティの高いスクリプトをだれでも簡単に記述することができるようになる。

7. 採択理由

近年 CI・DevOps が各開発運用現場に適用されているが、そこで使われているのはシェルスクリプトという古い言語のままである。bash や高級シェルを前提とする OS もあるが、Docker で使われている Alpine Linux ではバイナリサイズの小さい ash が標準で、シェルの持つ機能が少ない。シェルスクリプトは互換性を意識すると便利な拡張機能は使えず、プログラマにとっては非常にづらい書き方をしなければならない。JavaScript の Babel のように拡張性の高い文法で書いて、実行のバックエンドは古い互換性の高い言語にトランスパイルするような言語拡張環境をシェルスクリプトの世界で実現できることを期待して

いる。

8. 開発目標

シェルスクリプトは古い言語であり、一般的なプログラミング言語とは異なった構文を持っていることから記述するのが難しく、生産性を低下させており、バグの原因にもなっている。また、シェルスクリプトを使用しても、コマンドの使用方法などによってポータビリティを損なっていることも多く、適切な書き方をしなければ、高いポータビリティのスクリプトにはならない。シェルスクリプトで書かれていても、それほどポータビリティの高くないスクリプトも多いのが現状である。

このようなシェルスクリプトの弱点を克服するために、本プロジェクトでは現代的なプログラミング言語に近い文法と機能を持つ静的型付きスクリプト言語を開発する。この言語は、ポータビリティの高いシェルスクリプトへの変換を行うことで、書きやすい言語でありながら高いポータビリティを実現する。さらに、近年では多くの言語で採用されていながらシェルスクリプトでは利用できなかった静的型検査の手法を採用することで、より安全でバグの少ないスクリプトを書くことが可能となる。簡単に書くことができ、特段の配慮なしに高いポータビリティを得られる、これまでにない新しいスクリプト言語を普及させることを目指す。

9. 進捗概要

Cotowali のロゴタイプとロゴマークを制作した（図 1）。英文字フォント Montserrat をベースに独自のロゴタイプをデザインした。ロゴマークは、漢字の「理」とカタカナの「コトワリ」を組み合わせでデザインした。印章をイメージした角丸の形状と、筆文字をイメージしたカタカナの「コトワリ」の部分の切り欠きは、シェルスクリプトの長い歴史を、漢字や仮名文字の歴史に見立てたものである。



図 1 : Cotowali のロゴタイプとロゴマーク

本プロジェクトでは、以下のコマンドを開発した。

- lic : Cotowali コンパイラ
- lish : Cotowali 対話環境
- kuqi : 言語サーバ
- himorogi : Cotowali インタプリタ (PoC)

Cotowali コンパイラは V 言語で記述した。V 言語は 2019 年に登場した新しいプログラミング言語で、未成熟ではあるが一定程度動作する。V 言語は新しいプログラミング言語であるため、使用例が少ない。Cotowali は数万行単位のコードで構成されるが、この規模の V 言語で書かれたソフトウェアは、V 言語コミュニティが主導しているもの以外ではほとんど存在しない。Cotowali の開発過程では V 言語のバグに多く遭遇した。それらのバグは修正を行い、最終的には 50 を超える V 言語のバグが Cotowali の開発過程で修正された。

シンタックスハイライトを含むエディタプラグインを提供するために、より発展的なエディタサポート機能である言語サーバ Kuqi を開発した。これにより、Visual Studio Code や Vim などのソースコードエディタ上でエラー表示が行えるようになり、利便性が向上した。通常、エディタがプログラミング言語をサポートするためには、対象のプログラミング言語に対するサポートを実装する必要があるが、言語サーバを利用すると、エディタは言語サーバ仕様に対応するだけで、言語サーバを提供するすべてのプログラミング言語に対して機能を提供でき、プログラミング言語は言語サーバを提供するだけで、言語サーバをサポートする全てのエディタに機能を提供できた。

Cotowali で記述した FizzBuzz のソースコードを図 2 に示す。

```
fn fizzbuzz(i: int): string {
    if i % 3 == 0 && i % 5 == 0 {
        return 'fizzbuzz'
    } else if i % 3 == 0 {
        return 'fizz'
    } else if i % 5 == 0 {
        return 'buzz'
    } else {
        return "$i"
    }
}

for i in range(0, 20) {
    println(fizzbuzz(i + 1))
}
```

図 2 : Cotowali で記述した FizzBuzz のソースコード

10. プロジェクト評価

本プロジェクトでは、現代的なプログラミング言語に近い構文と機能を持つ静的型付きスクリプト言語 Cotowali を開発し、シェルスクリプトへの変換を行い、標準ライブラリが環境の差異を吸収することで、高いポータビリティを実現した。早くからオープンソースで公開しながら開発を進め、ドキュメントやシンタックスハイライト等の周辺環境も整備し、当初は予定していなかった Windows 環境に対応するため PowerShell バックエンドを開発するなど、PM の期待を超える開発成果を生み出した。本プロジェクト期間終了後に Hacker News に取り上げられた。GitHub スター数も 506 (2022 年 4 月 4 日時点) を超え、カザフスタンからもプルリクエストが飛んできている。世界レベルで話題になった。

11. 今後の課題

Cotowali を実際の業務プロジェクトへ適用することが今後の課題である。業務プロジェクトのセットアップやインストーラは複雑化する傾向にあり、未だにシェルスクリプトが使われている現場では、気が付いたらシェルスクリプトが想定通り動かなくなって事故につながることもある。ポータビリティを重視する Cotowali を利用すれば、その問題を解決できる可能性がある。今後の継続的な開発体制の維持構築のため、本プロジェクト期間終了後にクリエイターは Open Collective でスポンサー募集を開始した (<https://t.co/rNSnqDv16S>)。オープンソース開発者を支援する文化が日本でも根付くと良いと考える。