

C to Rust トランスパイラの試作

B175949 広島大学 *折岡砂璃 ORIOKA Sari

1. はじめに

家電機器や自動車などに組み込まれているコンピュータシステムは、社会において年々重要度が増している上、誤動作や異常などのトラブルによる影響が大きく、高い安全性と信頼性が要求されるものである。

現在、組み込み開発分野ではC言語による開発が主である。C言語による組み込み開発には、同言語で記述された既存資産が多いため互換性がとりやすく、処理速度が高速であるためリアルタイム性が要求される組み込みに最適であるという利点がある。しかし、同時にチェック機能脆弱性による安全性の問題や、メモリをプログラマが管理するため、メモリーリークやメモリの二重開放が発生する可能性を持つという欠点が存在する。

その欠点を補うため、MISRA-CというC言語のためのコーディング・ガイドラインが制定されている。これは全ての組み込みソフトウェアが対象であり、これらのガイドラインを守ることにより安全性をある程度保証することが可能であるというものである。欧州発であるが日本や北米でも広く普及しており、そのルールは1998年の初版から継続的に改訂がなされているため、極めて正確なガイドラインであるといえる。しかし、MISRA-C:2012には159件ものルール（必須ルール7件、必要ルール121件、推奨ルール31件）がある。これらのルールの多くは静的解析ツールにより適合・非適合の判定が可能であるが、コンパイル時にコードのチェックが可能であればより生産性は向上するだろう。

2. Rust について

そこで、Rustを次世代の組み込み開発言語として推奨したい。RustとはFirefoxを開発するMozillaが開発し、2015年に1.0版がリリースされた比較的新しいプログラミング言語である。特徴としては、所有権や借用システムといった独自のルールにより保たれる安全性、実行速度、並行性があげられる。従って、Rustによる組み込み開発の利点はコンパイル時の検証による安全性の向上、それに伴った安全性の向上、設計・プログラム品質の向上があげられる。独自のルールにより習得難度は高いが、コンパイル時に安全性についてある程度保証が可能であり、一度習得すれば生産性の向上も見込める。Rustへの開発言語の移行にあたって、

C言語で記述された既存のプログラムが多く存在するという点は無視できないものである。これらの既存資産をRustに自動的に変換できれば、Rustへの移行もより円滑に進められるだろう。そのため、MISRA-C準拠で記述されたC言語コードをRustに変換するツールの開発を研究内容とした。

3. トランスパイラの試作

先行研究においてC言語からRustへ変換するツールは既に存在するが、安全でないRustへの変換が可能なのである。本稿で議論するツールはRustの特徴を活用できるよう変換することを目的とし、必要に応じて手作業での介入も行うものとする。変換にはANTLRを用いる。ANTLRとは構文解析器を使用するためのパーサジェネレータであり、これを用いて作成された抽象構文木を元に、目的に合った解析を行うことができる。

4. まとめと今後の課題

本稿では、組み込み開発の現状とRust言語の特徴、Rustを生かした組み込みシステム開発への環境づくりについて議論した。今後は、MISRA-C準拠を前提としたC言語のコードからRustへの変換ルールを制定し、大まかな変換ルールが制定でき次第変換ツールの開発に移行する予定である。

References

- [1] keen, 河野達也, 小松礼人: 実践 Rust 入門, 技術評論社 (2019)
- [2] C2Rust-C 言語を Rust に変換,
<https://www.moongift.jp/2018/07/c2rust-c%E8%A8%80%E8%AA%9E%E3%82%92rust%E3%81%AB%E5%A4%89%E6%8F%9B/>,
2018/07/15