第1章 C言語プログラム歴45年の筆者が語る

ベテランC/C++プログラマからみたRustの安全性

山本 彰一

近年，Rustはいろいろな所で使われており，C/C++の代替となりえる，コード生成とメモリ安全性を持った新しい言語として注目を浴びています．

最初に筆者のプログラム歴を簡単に紹介します．C言語のプログラム歴45年，C++言語のプログラム歴30年，Rust言語のプログラム歴1年です．

C/C++のベテラン（多分現在までには，軽く100万行以上のコードを書いている）で，現在でもデバッガのコードを書いている現役プログラマです．

一方，Rustについては新米プログラマです．私の働く会社に入社してきたプログラマがRustに非常に詳しく，私の師匠です（年齢的には，私の息子より若い）．本稿は師匠の監修の元，執筆しています．

師匠曰く，「RustはC/C++の反省の結果，作られている」

この言葉に触発され，C/C++のプログラマを長年やってきた筆者が最近，Rustを学んだ結果，感じたことを記します．

● 2大特徴…安全性とモダン

私の思うRustの2大特徴は次の2つです．

1，メモリ安全性（空間的安全性，時間的安全性）をコンパイル時にチェックできる文法を持った言語
2，最近のモダン言語の特徴を持った言語

今回は，1のメモリ安全性についてC/C++プログラマとして感じたことについて説明します．

● C/C++と比べたときの特徴

言語仕様を眺めた段階では，C/C++とあまり変わりはありません（関数があり，引数があり，if，for，whileなども似たようなもの）．サンプル・コードをコンパイルするのも簡単です．

特徴的なのは，所有権，借用，ライフタイムです．

所有権：メモリをアクセスする権利．1つの変数に対して同時に書き込めないこと
借用：C/C++のポインタのようなものだが，アクセス範囲が必ず確定していなければならない
ライフタイム：変数の存在する期間で，ローカル変数のスコープ範囲やヒープ・メモリ変数のメモリを解放するまでの有効期間

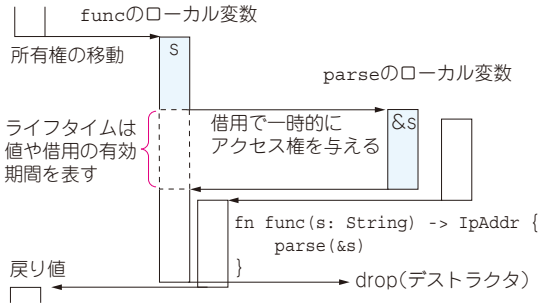

図1 Rustの所有権，借用，ライフタイムの関係

Rustの所有権，借用，ライフタイムの関係を図1に示します．

● Rustを学ぶなら「設計ポリシ」の理解が重要

どれも，C/C++のプログラムでも意識していたはずなのですが，いざRustで同じ動作をするプログラムを書こうとするとコンパイル・エラーとなり，コンパイルできない状態となってしまいました．一体何が問題なのでしょうか．

それは，Rust言語の設計ポリシをよく理解できていないためにそのような事態となってしまったのです．それから師匠に聞いたりネットで情報を集めたりを繰り返し，やっとRustでもコードが少しずつ書けるようになって来た段階です．

そこで，Rust初心者の経験を基に，「RustはC/C++の反省の結果作られている」を少しでも理解してもらえるよう簡単なC言語のサンプルとRustサンプルを挙げて説明していきます．

C/C++との比較①…空間的安全性

● 例題1…strlen関数

▶C/C++…正常動作の範囲外だと予測不能になる

リスト1(a)にC言語で文字列の長さを返すstrlen関数の最も簡単なバージョンを書いてみま