3 自動車業界のコーディング品質向上策 MISRA-C

――危険なC言語を安全に記述するためのガイドライン

黒田光洋

MISRA-Cは，英国の自動車関連ソフトウェアの業界団体であるThe Motor Industry Software Reliability Associationが策定した，C言語によるコーディングのガイドラインである．「コーディング規約」と聞くとめんどうな規則と思うかもしれないが，C言語を安全に利用するために必要なノウハウの一つである．しかし，MISRA-Cをそのまま導入すれば，すべての問題が解決されるわけではない．MISRA-Cのルールはあくまでも一つの見解，ととらえるのが妥当なようだ．（編集部）

1 C言語は危険がいっぱい

C言語は，そのシンプルな構造と自由度の高さから，組み込みソフトウェア開発やビジネス（IT）系アプリケーション・ソフトウェア開発などで広く使われています．しかし，そのC言語が危険な面を持った言語だということがどの程度認識されているのでしょうか？ C言語プログラミング経験の長いソフトウェア技術者でも，経験によって，危ないプログラミングの方法を「なんとなく」理解しているくらいではないかと思います．

C言語の規格書をまじめに読んだことのある人は多くないと思いますが，C言語の規格書（JIS X 3010-1993）[1]の巻末を見ると，付録「附属書G 可搬性」に，C言語では未規定の動作や未定義の動作，処理系（コンパイラや環境側）で定義する動作の一覧が示されています．「こういう場合にはどのように動作するかわかりません（保証できません）」，「こういう場合は，コンパイラによって動作や扱いが変わる可能性があります」といった項目が列挙されているのです．例えば，以下のようなものがあります．

- 静的記憶域の初期化の方法および時期は未規定
- 関数指示子および関数呼び出しの実引き数が評価される順序は未規定
- マクロ置き換え中に#および##の演算が評価される順序は未規定
- ビット・フィールドの型がint，signed intまたはunsigned intのいずれでもない場合は未定義
- 自動記憶域期間を持ち，初期化されていないオブジェクトの値を代入する前に使用している場合は未定義
- 関数が値を返していないのに，呼び出し元で関数呼び出しの値を使用している場合は未定義
- 負の値を持つ符号付き汎整数型の右シフトの結果は処理系が定義する
- switch文におけるcase値の最大数は処理系が定義する
- 単なるcharがsigned charと同じ値の範囲を持つか，unsigned charと同じ値の範囲を持つかは処理系が定義する

このような未規定の動作が22項目，未定義の動作が97項目，処理系が定義する動作が76項目，文化圏固有の動作（小数点文字，時間と日付の形式など）が6項目あります．

未規定または未定義の動作を実行してしまうと，プログラムが暴走したり，予期しない結果となる可能性があります．ただ，コンパイラによっては，独自にトラブルを回避するような対策を施しているものもあります（例えば，あらかじめメモリを初期化しておき，新しく宣言された変数に初期値として0が入るようにしておくなど）．そのような場合，これまで問題なく動いていても，開発環境やコンパイラを変えたとたんに動作がおかしくなった，という状況になることがあります．

C言語は便利で融通が利く反面，危険な側面も持ってい

ることを，プログラムを書く人は認識する必要があります．

● 自動車業界の品質意識から生まれたMISRA-C

MISRA-Cというのは，英国の自動車関連ソフトウェアの業界団体であるThe Motor Industry Software Reliability Association（MISRA；「ミスラ」とか「ミズラ」と呼ばれている）が策定した，C言語によるコーディングのガイドラインです．MISRA-Cの正式名称は「Guidelines for the Use of the C Language in Vehicle Based Software」[2]であり，冊子として販売されています（日本語訳は自動車技術会が販売している[3]）．

MISRA-Cは，自由度が高いC言語を安全に使うため，「こういう記述はよくないのでは？」という項目を「ルール（規約）」としてまとめたガイドラインです．コーディング規約であるとも言えますが，準拠していなければ製品を納入できない，などの拘束力はありません．あくまでもガイドラインという位置づけです．MISRA-Cは規格上の問題を回避するためだけではなく，過去の経験や移植性，可読性についても考慮されたガイドラインで，バランスのとれた内容であると筆者は感じています．

C言語のコーディング規約を社内で規定している企業もあるようですが，ひと口にコーディング規約と言っても，規格書をもとに作成されたもの，経験を積み上げたもの，スタイルを統一するためのものなどさまざまです．それらはいずれも企業の経験やノウハウの蓄積であり，無償で一般に広く公開されることはありません．MISRA-Cは，自動車業界全体のソフトウェアの品質を向上させるという目的のために，貴重な情報を惜しげもなく公開したものと見ることもできます．

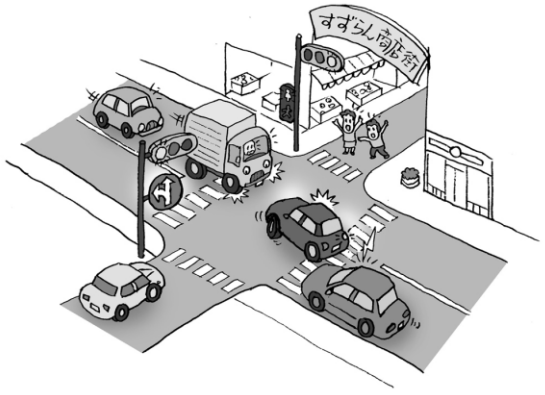

図1　なぜ右折禁止なのか理由を知る

● 自動車以外の業界も採用

1998年4月にMISRA-Cが公開されてから，英国では自動車以外の業界でも多く利用されています．最近はMISRA-Cが広く知られるようになり，日本をはじめドイツ，米国など，世界中で利用されつつあります．最近話題になった中国の有人ロケットの開発でもMISRA-Cが利用されたようです．

日本では，日本語訳[3]が出版された2002年以降，MISRA-Cに関心を持つ企業が増えてきました[注1]．一部の自動車メーカなどは，2002年以前からMISRA-Cに注目し，研究していたようです．

MISRA-Cは，組み込みソフトウェア開発はもちろん，ビジネス（IT）系のソフトウェア開発にも十分利用できるガイドラインだと筆者は考えています．今後も業界を問わず，さらに幅広く利用されるのではないかと予測しています．

● 押しつけではない，危険を避けるためのアドバイス

「コーディング規約」と聞くと，めんどうだとか，うっとうしい規則だという印象もあるでしょう．ここでプログラミングを，自動車の運転に例えて考えてみましょう．自動車を運転する前には，まず操作や交通規則などを学びます．自動車教習所に通うと，標識の意味をうるさく教えられたりします．そのとき重要なことは，例えば「右折禁止」の道は，なぜ右折禁止なのかを理解することです（「その標識があるから」右折禁止であるわけではない）．そのために，「右折禁止の道で右折したら，車が向こうからやってきて事故になってしまった」という映像を見せられたりします（**図1**）．

交通規則は，複雑な交通社会を安全に乗り切るための知恵です．コーディング規約も同じで，単に「やってはいけない」ではなく，「やると危険だ」と教えてくれる知恵なのです．だから，「なぜ，その規約が存在するのか」を理解してコーディングすることが重要です[注2]．

注1：筆者の所属するMISRA-C研究会も，2002年4月に発足した．MISRA-C研究会は，MISRA-Cを日本で利用するためにどう解釈するかを現場の視点で考える研究会である．自動車関連ソフトウェアを開発する企業や，ツール・ベンダなどを含む13社によって発足した（2004年6月現在の参加企業は27社）．MISRA-Cについて自由に議論できるように，業界に対する影響力の大きい自動車メーカにはあえて参加を遠慮してもらっている．現在はSESSAME（組込みソフトウェア管理者・技術者育成研究会）のワーキング・グループとして活動している．2004年5月にはMISRA-Cの解説書[6]を編集，日本規格協会より出版し，「日本でまじめにMISRA-Cに取り組んでいる団体」として認知されている．

注2：実際には，しごとに追われている現場の開発者に「理解しろ」と言っても，なかなか重要性を認識してもらえないのが現状である．だからこそ継続的な教育がたいせつ，と新入社員教育や研修の重要性を身にしみて感じている今日このごろである．