

MISRA C 2025：新機能の理由と背景

2025.3.10

セーフティクリティカルなソフトウェアや高いセキュリティのソフトウェアを扱っている方なら、システムの堅牢性や信頼性、安全性を維持するために設計されたコーディングガイドライン集である MISRA C をご存じでしょう。1998 年に初導入されて以来 MISRA C は、航空宇宙、防衛、自動車、医療、産業オートメーションなど、コード品質が絶対である業界で鉄板の標準となっています。

このガイドラインは長年にわたって、組込みシステムやソフトウェア集約型システムでの新たな課題に対応するために進化してきました。最新のアップデートでもその役割は引き継がれ、開発者が安全でセキュリティが高く、保守しやすい C コードを容易に作成できるようにしています。もし LDRA ツールスイートをご使用でしたら嬉しいことに、規格として正式リリースされた日に、この最新の MISRA C 改訂版のサポートが利用可能になっています。



ソフトウェアの脆弱性に正面から取り組む

ソフトウェアの脆弱性、特にメモリの安全性の問題は、開発者と OEM (Original Equipment Manufacturer) 双方にとって大きな悩みの種です。しかし、LDRA の技術スペシャリストであり、MISRA C ワーキンググループ議長である Andrew Banks 氏が指摘するように、これらの脆弱性の多くは C 言語自体に起因するものではありません。むしろ、開発者が手抜きをしたり、ベストプラクティスを無視したりすることで発生することが多いものです。



Andrew Banks 氏

MISRA Cに従うことで、開発者がメモリ安全性での落とし穴を回避し、コードの明確さを向上させ、テストを効率化して、ソフトウェアが長期的に保守可能であることを保証できます。つまり、このガイドラインはコードをより安全にするだけでなく、作業をより快適にします。

名前ではなく内容です

MISRA C:2023は MISRA C:2012 の直接の後継版に見えます。しかし、2019年に MISRA C:2012 (これは最初の MISRA-C:1098 からみて第3版) の改訂初版で大幅なアップデートが行われて状況が少し混乱しました。後から考えてみると、これを MISRA C:2019 と呼んだ方がわかりやすかったかもしれません (2023年の改訂で MISRA C:2023 – 第3版の改訂2版 – となったものです)。

したがって、名前に新しい年が含まれているだけで MISRA C:2025 が大規模な見直しを受けたものであるとは考えないでください。ここでの変更は、画期的な変化を示したというよりは業界からのフィードバックに対する MISRA 委員会の対応を反映したものです。

改訂2版とはいうものの、MISRA C:2023 は大きな変革をもたらしました。マルチスレッドのサポート、より安全な並行動作のためのアトミック型、より統一された参照構造、が導入されて以前の版が単一の包括的な版に統合されました。

一方、MISRA C:2025 は、より微調整されたアップデートです。強化されたポリシーと追加のガイドラインを導入して、コンプライアンスがスムーズになり、将来の C 言語規格改訂の準備が整いました。

フロントマテリアルの再構成

MISRA C 規格文書の「フロントマテリアル」は、コアである技術内容の前に置かれる導入セクションです。それらのセクションは、その技術内容の範囲や意図、構造を理解するのに役立ちます。

ここでの変更のほとんどは、セクション6「ガイドラインの概要」に影響を与えていますが、適用解除のガイドラインを扱う新規のサブセクション6.4.4 以外は、順序と番号付けの変更が主です。

MISRA C ガイドラインの変更

新しいガイドラインでは 79 件の変更があり、その分類は次のようになります。

新しいガイドライン	4 件
番号が変更されたガイドライン	3 件
削除されたガイドライン	2 件
適用解除のガイドライン	1 件
その他の変更されたガイドライン	69 件
規範的または技術的な変更 (コンプライアンスの遵守に影響を与える拘束力のあるルール、ディレクティブ、要件)	12 件
情報および編集上の変更	57 件

また、純粋に編集上の変更が2件あります。

ガイドラインの変更

4件の新しいガイドラインについて、それらが導入された理由を簡単に見てみます。

- R.8.18 (必須) ヘッダーファイルでは仮定義を行ってはならない

仮定義は、明示的な初期化子なしで変数が宣言されたときに発生します。仮定義が複数の翻訳単位(つまり、同じヘッダーを含む異なる.cファイル)に出現すると、リンカエラーや意図しない複数の定義が発生する可能性があります。

- R.8.19 (推奨) ソースファイルでは外部宣言を行ってはならない

.cファイルで外部(extern)宣言を使用することは、保守性の低下や隠れた依存関係、潜在的なリンクでの問題につながる可能性があるため、一般的には悪い習慣と見なされます。

- R.11.11 (推奨) ポインタは暗黙的に NULL と比較してはならない

ポインタを暗黙的に NULL と比較する(つまり、明示的な比較を省略する)と、コードの明確さが低下し、混乱が生じ、意図しない論理エラーが発生する可能性があります。明示的な比較を記述すると、可読性が向上し誤解が減ります。

- R.19.3 (必須) 共用体メンバは、以前に設定されていない限り読出してはならない

共用体メンバに明示的に値が割り当てられる前に読み出すことは、Cでは未定義の動作です。これにより、ガベージ値、クラッシュ、意図しないプログラム動作など、予測できない結果が発生する可能性があります。

残りのガイドライン変更は主に改良であり、1つのルールがディレクティブに変更、2つのガイドラインがそれぞれ他のガイドラインと内容統合されて削除、1つのガイドラインが適用解除になりました。

例題の見直し

ポリシーの変更によって、いくつかの例題が更新されました。特に次の点になります。

- ディレクティブ 4.6 では、「基本数値型に替えて、サイズと符号属性を示す typedef を使用しなければならない」と規定されています。浮動小数点型 (Complex を含む) は、本質的に処理系定義であり、その特性はサイズだけにとどまらないため、このディレクティブの適用範囲から削除されました。
- 以前は一部の例題に、NULL の定義を取得するため `#include <stdlib.h>` が含まれていました。これらは、処理系間で普遍的に使用できる `#include <stddef.h>` に変更されました。

付録に対する変更

付録のアップデートは、フロントマテリアルに対する変更とほとんど同様のものです。これには、表の小規模な変更と参照内容を容易に確認できるようにするためのサブセクション識別子の導入が含まれます。もう1つの注目すべき変更は、付録I(逸脱記録例のプレースホルダー)の削除です。

参照する用語体系の見直し

MISRA ガイドラインドキュメントでは、「参照文献」は MISRA ガイドラインが引用または依存している外部ドキュメントや規格、情報源を指します。MISRA C:2025 では、C90 の処理系定義の動作が C 言語規格との一貫性を保ち、未定義および未指定の動作をより明確に注釈付けできるように、異なる方法で列挙されています。

まとめ

新しい MISRA C:2025 ドキュメントは、MISRA C:2023 の効率性に基づいて構築され、意味のあるポリシー変更、機能強化、そして安全でセキュアなコードの開発を効率化するための新しいガイドラインを導入しています。さらに重要なのは、C 規格の将来の更新に対する基盤を確立し、セーフティクリティカルなアプリケーションにおける継続的なコンプライアンスと堅牢性を保証することです。

大規模な見直しではないかもしれませんが、このアップデートは、C 言語でのコーディングをより安全で明確で保守しやすいものにするための継続的な取り組みを反映したものです。コンプライアンス作業に深く関わっている場合でも、より良いコードの作成を目指しているだけの場合でも、これらの改良は注目に値します。

MISRA C:2025 の適用と LDRA でのサポートに関する詳細は、以下をご覧ください。

[Blog: MISRA Myths Busted #1 – Is MISRA C *just* an Automotive Standard?](#)

[Blog: MISRA Myths Busted #2 – Is MISRA C *just* a Safety Standard?](#)

[Website: The MISRA Language Guidelines, MISRA C, and MISRA C++ – What they're for, what's current and what isn't.](#)

[Website: Coding standards compliance for embedded systems](#)

[Website: Standards development](#)



富士設備工業株式会社 電子機器事業部 www.fuji-setsu.co.jp