

≫ 文法の曖昧さを理解して確実性と再利用性を高める

マイコンC言語 転ばぬ先のつえ

第15回 演算子⑥…関係演算子と等値演算子

鹿取 祐二

本連載では、C言語の言語仕様(文法)の曖昧な部分と、それにより起こる問題を解説します。再利用性と効率が高く、安全かつ安心して使えるソフトウェアが開発できるようになることを目指します。

今回は、関係演算子と等値演算子を使うときの注意事項について解説します。 (編集部)

24 比較結果を利用して条件分岐を効率よく実行できる場合もある

● 比較結果が他の演算にも利用できることに注意

大きい「>」、小さい「<」、以上「>=」、以下「<=」、等しい「==」、等しくない「!=」、各種の制御文で比較を行う際に利用される関係演算子と等値演算子ですが、これらは比較だけを行う演算子ではありません。比較した結果を0と1で表現するのが正しい演算子の動作です。

C言語では0でなければ真、0ならば偽となるので、比較した結果が真ならば1、偽ならば0が、これら演算子の評価結果です。しかも、その1と0はint型であることも定められています。当然ながら、型が決まっているのですから、それは他の演算に利用できます。

そう言われてもピンと来ない方もいらっしゃるかと思います。先程の内容は、多くの入門書では次のように説明されています。

```
if ( a < b < 2 ) // 常に真となる比較
```

上記は、aはbより小さく、なおかつ、bは2より小さい、とはなりません。紹介した通り、比較した結果は1か0のどちらかです。つまり、a < bの結果は1か0のどちらかですから、次に< 2と比較すれば、どちらに転んだとしても2よりは小さいため、aとbに関係なく、必ず真となります。

● if文のような使い方も可能

前述の使い方は一般にNGの例として紹介されており、それを逆に利用しようと思う人は少ないでしょう。MISRA-Cにも比較結果を他の演算に利用してはならない、といったルールは存在しません。詳細は定かではありませんが、今から紹介する記述が可能であることが理

解されていないから、ルールが存在しないのかもしれませんが。

```
int a[2], b, c;
a[ b == c ] = 10 << ( b == c );
```

特に意味がある式ではありませんが、理解できるでしょうか。上記を分かりやすい記述に直すと次のようになります。

```
int a[2], b, c;
if ( b == c )
    a[1] = 20;
else
    a[0] = 10;
```

比較結果は1か0のどちらかです。ゆっくりと考えれば、両者が同じ結果となることが理解できると思います。

● if文と比べたときの性能評価

他の演算に利用できる比較結果ですが、性能はどのようなのでしょうか。文法的に利用できても、性能が悪い、またはif文と変わらないのであれば利用する価値がありません。次の簡単な例で性能を見てみましょう。

```
int a, b, c;
c = a == b;
```

上記のコンパイル結果をマイコンごとに示します。

▶ 性能の評価方法…分岐命令があるかないかで評価

性能の良し悪しは分岐命令の存在で分かります。真と偽のそれぞれに文が存在するif文の場合、必ず分岐命令が必要となります。一方、比較結果をそのまま式として利用する場合、処理の流れは1つですから、基本的に分岐命令は必要ありません。

問題は、それを各マイコンが実現できるかどうか、もっと正確に言えば比較結果を汎用レジスタなどに読み出す命令を持っているかどうかです。もし、そのような命令を持っていない場合、if文と同じように分岐命令を使って処理を実現することになるので、比較結果を式として利用する意味がなくなってしまいます。

▶ 分岐命令がある…H8やRL78の場合

H8やRL78の場合、どちらにも分岐命令が存在して

第6回 派生型①…難しい配列やポインタをtypedefで分かりやすくする (2021年7月号)

第7回 派生型②…構造体を使うときの作法とコード・サイズ削減テクニック (2021年9月号)

第8回 派生型③…ちゃんと思えばメモリ容量の削減に/共用体とは (2021年10月号)