

M3T-NC308WA, M3T-NC30WA ご使用上のお願い

Cコンパイラ(アセンブラ・統合化開発環境付き) M3T-NC308WA, M3T-NC30WA の使用上の注意事項を連絡します。

- if文の条件が真の場合と偽の場合とで同一の関数を呼び出す場合の注意事項

1. 該当製品

M32C/80シリーズ, M16C/80, およびM16C/70シリーズ用 :
M3T-NC308WA V.5.00 Release 1 ~ V.5.20 Release 1

M16C/60, M16C/30, M16C/20, M16C/10, R8C/Tinyシリーズ用 :
M3T-NC30WA V.5.00 Release 1 ~ V.5.30 Release 1

2. 内容

if文の条件が成立する場合に実行される副文と、条件が成立しない場合に実行される副文の両方で同じ関数を同じ引数で呼び出している場合に、その関数の戻り値が不正になることがあります。

2.1 発生条件

以下の条件をすべて満たす場合に発生します。

- (1) コンパイル時に最適化オプション -OR を指定している。
- (2) 自動変数または呼び出し元の仮引数と、定数0(ゼロ)を比較する制御式を持つif文が存在する。
- (3) (2)の変数または仮引数の型が、char, signed char, unsigned char, int, unsigned int, short, またはunsigned short のいずれかである。

※1. 変数または仮引数がvolatile修飾されているかどうかは無関係に発生する。

※2. 変数または仮引数にアドレス演算子を使用していても発

生する。

- (4) (2)のif文の条件が成立する場合に実行される副文Aと、条件が成立しない場合に実行される副文Bの両方で、同一の関数Cを同一の引数で呼び出している。
- (5) 関数Cの戻り値が整数型、浮動小数点数型、またはポインタ型である。
- (6) 関数Cは、スタック渡しの引数を持たない。
- (7) 副文Aと副文Bの両方で条件分岐直後から関数Cが呼び出されるまでの処理が同一である。

2.2 発生例

```
-----char sub(void);          // 発生条件
(5),(6)char func(int dummy, char c){  if (c == 0) {          // 発生条件(2), (3)
return sub(); // 発生条件(4), (7) }  sub();          // 発生条件(4), (7) return
0;}-----
```

3. 回避策

if文で条件が成立する場合の処理と条件が成立しない場合の処理のいずれか一方の先頭にダミーのasm関数を挿入してください。

```
-----char sub(void);char func(int dummy,
char c){  if (c == 0) {          asm();          // ダミーの asm()          return sub(); }
sub();  return 0;}-----
```

4. 恒久対策

次期バージョンで改修する予定です。

[免責事項]

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。ニュース本文中のURLを予告なしに変更または中止することがありますので、あらかじめご承知ください。