

**Cコンパイラパッケージ
M3T-NC308WAおよびM3T-NC30WA
ご使用上のお願い
--最適化オプション"-OR"使用時の注意事項--**

M16Cファミリ用CコンパイラパッケージM3T-NC308WAおよびM3T-NC30WAの使用上の注意事項を連絡します。

- 最適化オプション"-OR"使用時のアプリケーションエラーについて

1. 該当製品

M32C/90, M32C/80, M16C/80および M16C/70シリーズ用:
M3T-NC308WA V.5.00 Release 1 ~ V.5.20 Release 02

M16C/60, M16C/30, M16C/20, M16C/10, M16C/Tinyおよび R8C/Tinyシリーズ用:
M3T-NC30WA V.5.00 Release 1 ~ V.5.30 Release 02

2. 内容

最適化オプション"-OR"を選択して、関数呼び出しを含むソースをコンパイルしたとき、Windows版ではアプリケーションエラーが、EWS版およびLinux版ではコアダンプが発生し、コンパイルが強制終了される場合があります。

注意：

後述の発生条件に該当してもコンパイルが正常に終了する場合があります。その場合に生成されたコードは、本件による問題は ありません。

2.1 発生条件

以下の条件をすべて満たす場合に発生することがあります。

- (1) 最適化オプション"-OR"が選択されている。
- (2) 以下のいずれかに該当するような複数の関数呼び出し箇所があ

る。

- a. 1つの関数内の複数の箇所で、同じ関数を呼び出している。
- b. 1つの関数内の複数の箇所で、同じポインタを介して、間接的に関数を呼び出している。
- c. 1つの関数内の複数の箇所で、同じ共用体に属するポインタを介して、間接的に関数を呼び出している。

注: b.およびc.は、複数の個所で同じ関数を呼んでいても、呼び出し箇所ごとに異なる関数を呼んでいても、いずれも該当します。

(3) (2)の関数呼び出しのうち、2つ以上の呼び出し後の処理が合流する。

(4) (3)の合流箇所の前のそれぞれの関数呼び出しの中に、レジスタを使って渡される実引数の個数が異なるものがある。

注意:

発生条件 (2) の a. および b. は、ソースに誤りがある場合です。

2.2 発生例

発生条件(2)c.に該当する例

(同じ共用体に属するポインタで間接呼び出ししている例)

```
-----  
union {  
    void (*pf0)(void);  
    void (*pf1)(int);  
} u;  
  
int example2(int i)  
{  
    switch (i) {  
        case 0:  
        case 2:  
            (*u.pf1)(0); /* 発生条件(2)c.および (4) */  
            break;  
        default:  
            (*u.pf0)(); /* 発生条件(2)c.および (4) */  
            break;  
    }  
    return 0; /* 発生条件(3) */  
}
```

発生条件(2)a.に該当する例

(同じ関数を複数個所で呼び出している例)

注意：

この例は、ソースの誤記のために、発生条件に該当する例です。"else"側の関数呼び出しの実引数の個数が、プロトタイプ宣言と一致していません。これに類似したソースをコンパイルすると、コンパイルが正常終了したとしても、警告メッセージ"too few parameters"が出力されます。

void func(int);

```
int example1(int i)
{
    if (i == 0) {
        func(0);    /* 発生条件(2)a, (3), (4) */
    } else {
        func();    /* 発生条件(2)a, (3), (4) */
    }
    return 0;    /* 発生条件(3) */
}
```

3. 回避策

以下のいずれかで回避してください。

- 同じ関数を呼び出している場合、または、同じポインタを介して間接的に関数を呼び出ししている場合は、実引数の個数がプロトタイプ宣言に合致するよう、関数の実引数を修正してください。
- 同じ共用体に属するポインタを介して間接的に関数を呼び出している場合には、処理が合流する前の関数呼び出しのいずれかの直後に、ダミーのasm関数を挿入してください。

発生例の回避例

union {
 void (*pf0)(void);
 void (*pf1)(int);
} u;

```
int example2(int i)
{
    switch (i) {
```

```
case 0:
case 2:
    (*u.pf1)(0);
    break;
default:
    (*u.pf0)();
    asm();    /* ここに asm(); を挿入してください。 */
    break;
}
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。