RENESAS ツールニュース

RENESAS TOOL NEWS 2013年11月16日：131116/tn1

CubeSuite+ CC-RXコンパイラ ご使用上のお願い

CubeSuite+ CC-RXコンパイラの使用上の注意事項を連絡します。

- 構造体変数の格納領域が正しく確保されない場合の注意事項 (RXC#025)

注: 注意事項の後ろの番号は、注意事項の識別番号です。

---

1. 該当製品

CubeSuite+ CC-RXコンパイラ V2.00.00 および V2.00.01

2. 内容

コンパイラの最適化によって、初期値が設定されている構造体変数の格納用に
確保される領域が、必要なサイズより小さくなり、正しく領域が確保されない
コードを生成する場合があります。
その場合、該当する構造体を操作するプログラムが正しく実行されません。

3. 発生条件

以下の条件を全て満たす場合に発生することがあります。
(1) 共用体をメンバに持つ構造体が存在する。
(2) (1)の構造体のメンバである共用体とその直前のメンバの間にパディングが入る。 (注1)
(3) (1)の構造体のメンバである共用体のメンバのアライメント数は一致していない。 (注2)
(4) (1)の構造体のメンバである共用体に初期値が設定されている。
(5) 初期値が設定されている構造体のメンバのアライメント数は、共用体の
アライメント数よりも小さい。

注1: 直前のメンバの領域の終了位置が共用体のアライメント数 (境界調整数) と
合っていない場合、アライン (境界調整) のためにパディングが入ります。
注2: 基本型のサイズが1または2 バイトであるメンバを1つも含まない構造体の
場合、アライメント数は全て4で一致するので、条件(3)は成立せず、
本問題には該当しません。

発生例:

```
-----------------------------------------------------------------------
typedef struct {
   char x;   // 共用体の直前のメンバ (アライメント数は1)
             // 共用体U1の直前のメンバxの領域の終了位置は
             // 構造体の先頭から + 1 バイトの位置になる。
             // このため、xとU1の間には3 バイト分のパディングが
             // 入る。  発生条件(2)
   union {       // 共用体のアライメント数は4   発生条件(1)
      char m;    // メンバmのアライメント数は1  発生条件(3)
      long n;    // メンバnのアライメント数は4  発生条件(3)
   } U1;
} Str;

Str S1 = {
   0x77,
   {
      0x88  // 共用体の初期値を、共用体のアライメント数4よりも
   }        // アライメント数が小さいメンバmに対して設定
};          // している。  発生条件(4)および(5)
-----------------------------------------------------------------------
```

上記の記述では、構造体S1においてメンバxの後にパディングが出力されなくなり、メンバU1に対応する後半の4 バイトが正しいアドレスに配置されません。

4. 回避策

以下のいずれかの方法で回避してください。

(1) 当該共用体の直前に、パディング代わりのダミーの変数を追加する。

例:

```
-----------------------------------------------------------------
typedef struct {
   char  x;
   char  dummy1; // 3 バイト分のパディングの代わりとなるダミーの
   short dummy2; // 変数を追加する。
   union {
      char m;
      long n;
   } U1;
} Str;
-----------------------------------------------------------------
```

(2) packオプションまたは#pragma pack (注) の機能を利用して当該構造体内のメンバのアライメント数を1にする。

注: 構造体がauto変数の場合は、#pragma packは適用できません。

(3) 構造体内のメンバの順序を入れ替え、当該共用体の直前のパディングが

不要となる位置に変更する。
発生条件(2)を満たす共用体が親構造体内に1つの場合は、その共用体を
先頭メンバに移動する。
例:

```
----------------------------------------------------------------
typedef struct {
   union {      // 直前にパディングが入らない位置に共用体の宣言を
      char m;   // 移動する。
      long n;   // (構造体の先頭であればアラインは不要である)
   } U1;
   char x;
} Str;
----------------------------------------------------------------
```

(4) 発生条件(4)の初期値を共用体内のアライメント数が最大のメンバに
対して設定するように変更する
(a) C89およびC++の場合 (lang=c99を選択せずコンパイルする場合)
例:

```
----------------------------------------------------------------
typedef struct {
   char  x;
   union {
      long n;  // 共用体内でアライメント数が最大のメンバを先頭
      char m;  //に宣言し、初期値の設定先となるようにする。
   } U1;
} Str;

Str S1 = {
   0x77,
   {
      0x88     // 注意: ビッグエンディアンの場合は初期値を
   }           //       0x88000000とする必要があります。
};
----------------------------------------------------------------
```

(b) C99の場合 (lang=c99を選択してコンパイルする場合)
例:

```
----------------------------------------------------------------
typedef struct {
   char  x;
   union {
      char m;
      long n;
   } U1;
} Str;
```

```
Str S1 = {
   0x77,
   {
      .n = 0x88  // 共用体内でアライメント数が最大のメンバの
                 // 初期化式になるように、メンバ名付きの初期化
                 // とする。
   }             // 注意: ビッグエンディアンの場合は初期値を
};               //       0x88000000とする必要があります。
```

---

5. 恒久対策

CubeSuite+ CC-RXコンパイラ V2.01.00で改修しました。
V2.01.00の詳細は、RENESAS TOOL NEWS 資料番号131116/tn2 を
参照ください。以下のURLでも参照できます。 (11月19日から公開予定)
https://www.renesas.com/search/keyword-search.html#genre=document&q=131116tn2
