

M32Rファミリ用C/C++コンパイラパッケージ V.5.01 Release 00 ご使用上のお願い --1万個を超える初期化式を記述した場合の注意事項--

M32Rファミリ用 C/C++コンパイラパッケージ M3T-CC32R V.5.01 Release 00 の 使用上の注意事項を
連絡します。

- 1万個を超える初期化式を記述した場合の注意事項

1. 内容

C/C++ソースコード中で、{ と } で囲まれた初期化式並びの中に、1万～5万程度以上の初期化式がある場合、M3T-CC32Rが次のようなメッセージのWindowsエラーを発生し、コンパイルできないことがあります。

Windowsエラーのメッセージ

問題が発生したため、postpar.exe を終了します。 ご不便をおかけして
申し訳ありません。

2. 発生条件

C++言語、C言語で発生条件が異なります。

以下のそれぞれの条件(1)(2)を共に満たす場合、本問題が発生する場合があります。

2.1 C++言語の場合

(1) short型またはchar型変数(unsignedやsignedが付加される場合を含みます)の配列宣言に対し、初期化式を約1万～5万個以上含む初期化式並びを記述している。

(2) 上記(1)の初期化式並びの前後の行に、別の宣言や関数の記述がある。

2.2 C言語の場合

- (1) 任意型の配列宣言に対し、キャスト付きの初期化式を約1万～5万個以上含む初期化式並びを記述している。
- (2) 次の(a)(b)いずれかの条件を満たす
 - (a) 上記(1)の配列宣言において配列個数が省略されている。
 - (b) 上記(1)の初期化式並びの後(下方)の行に、別の宣言や関数の記述がある。

3. 発生例

ソースファイル sample1.cpp

```
unsigned short table1[] = {      /* 発生条件2.1(1) */
    0,
    112,
    93,
    :
    /* 3万個、同様の記述が続く */
    :
};
int
check_table1(int index)        /* 発生条件2.1(2) */
{
    int ans;
    ans = table1[index];
    return ans;
}
```

ソースファイル sample2.c

```
int table2[] = {                /* 発生条件2.2(1),(2)(a) */
    (unsigned) 0,
    (unsigned) 100392,
    (unsigned) 33293,
    :
    /* 5万個、同様の記述が続く */
    :
};
```

ソースファイル sample3.c

```
int table3[50003] = {          /* 発生条件2.2(1) */
    (unsigned) 0,
    (unsigned) 100392,
    (unsigned) 33293,
    :
```

```

/* 以下、5万個、同様の記述が続く */
:
};
int
check_table3(int index)      /* 発生条件2.2(2)(b) */
{
    int ans;
    ans = table3[index];
    return ans;
}

```

コマンドライン例

```

% cc32R -S sample1.cpp
% cc32R -S sample2.c
% cc32R -S sample3.c
注: “%” は、プロンプトを表します。
-----

```

4. 回避策

次のいずれかの方法で回避ください。

(1) 該当する初期化式並びを含む宣言を別のC++ソースに移動する(C++言語の場合)

ソースファイル sample1.cpp の回避例

元の初期化式並びを含む宣言を、例えば、sample1init.cpp という新規のC++ソースファイルに移動してください。さらに、元の宣言からは初期化式を削除し、extern宣言を付けてください。

sample1init.cpp (新規作成)

```

-----
unsigned short table1[] = {
    0,
    112,
    93,
    :
    /* 3万個、同様の記述が続く */
    :
};
-----

```

sample1.cpp (変更後)

```

-----
extern unsigned short table1[]; /* 初期化式のないextern宣言に変更 */
int
check_table1(int index)
{

```

```
int ans;
ans = table1[index];
return ans;
}
```

(2) 該当初期化式のキャストを削除する(C言語の場合)

ソースファイル sample2.c の回避例

初期化式に記述したキャストを削除してください。

```
int table2[] = {
    0,                /* キャストを削除 */
    100392,           /* (以下同様) */
    33293,
    :
    /* 5万個、同様の記述が続く */
    :
};
```

(3) 当該宣言の配列個数の記述とソースコード末尾への移動(C言語の場合)

上記回避策の(2)が使用できない場合の代替手段です。

なお、発生条件2.2(2)の(a)と(b)両方に該当する場合は、下記の sample2.c と sample3.c で示す対策を共に実施する必要があります。

ソースファイル sample2.c の回避例

配列 table2 の個数を省略せずに記述してください。

```
int table2[50003] = {          /* 個数を記述するように変更 */
    (unsigned) 0,
    (unsigned) 100392,
    (unsigned) 33293,
    :
    /* 5万個、同様の記述が続く */
    :
};
```

ソースファイル sample3.c の回避例

配列 table3 の初期化つき宣言を関数の後(下方)の行に移動し、元の場所には初期化式並びのない宣言を記述してください。

```
int table3[50003];           /* 宣言のみを残す */
int
check_table3(int index)
{
    int ans;
```

```
ans = table3[index];
return ans;
}
int table3[50003] = {          /* ここに移動する */
    (unsigned) 0,
    (unsigned) 100392,
    (unsigned) 33293,
    :
    /* 5万個、同様の記述が続く */
    :
};
```

[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。

© 2010-2016 Renesas Electronics Corporation. All rights reserved.