

CS+ CXコンパイラパッケージのご使用上のお願い

CS+ CXコンパイラ (V850E2M/E2Sコア用) の使用上の注意事項を連絡します。

1. 1024バイト以上の整数型配列の自動変数を初期化する場合の注意事項(No.17)
2. pow関数の戻り値が不正となる注意事項(No.18)

注: 各注意事項の後ろの番号は、注意事項の識別番号です。

1.1024バイト以上の整数型配列の自動変数を初期化する場合の注意事項(No.17)

1.1 該当製品

CX V1.00~V1.31

1.2 内容

1024バイト以上の整数型配列の自動変数を初期化するとき、初期化子の数が要素数より少ない場合、暗黙的に0で初期化される部分の初期値が不定になることがあります。

1.3 発生条件

以下の条件をすべて満たす時に発生する場合があります。

- (1) 1024バイト以上の整数型配列を自動変数として宣言している。
- (2) (1)の配列の要素数より少ない数の初期化子で明示的に初期化している。
- (3) (2)の初期化子は文字列リテラル以外である。

発生例:

```
-----  
void func( void )  
{  
    char array_ng[1024] = {0};  
}
```

1.4 回避策

以下のいずれかの方法で回避してください。

- (1) 対応する初期化子がない要素に対して、ループを用いて明示的に0を代入する。
- (2) 対応する初期化子がない要素に対して、memset関数を用いて0を代入する。

回避策(1)の適用例:

```
-----  
void func( void )  
{  
    char array_ng[1024] = {0};  
    int i;  
    for (i = 1; i < 1024; ++i) {  
        array_ng[i] = 0;  
    }  
}
```

1.5 恒久対策

今後のバージョンで改修予定です。(次のリリース時期は未定です)

2. pow関数の戻り値が不正となる注意事項(No.18)

2.1 該当製品

CX V1.00~V1.31

2.2 内容

pow関数を使用してべき乗を計算する際、第1引数が負であり、かつ第2引数が2147483649 以上 4294967295 以下 または -4294967295 以上 -2147483649 以下の範囲内の奇数の整数値である場合、戻り値の符号が不当に正になります。

2.3 発生条件

以下の条件をすべて満たす時に発生します。

- (1) -Cオプションで指定したターゲットデバイスがFPUを持たない、または-Cオプションで指定したターゲットデバイスがFPUを持ち-Xfloat=softオプションを指定している。
- (2) pow関数の第1引数が負数である。
- (3) pow関数の第2引数が以下のいずれかの範囲の奇数である。
 - (a) 2147483649 以上 4294967295 以下
 - (b) -4294967295 以上 -2147483649 以下

ターゲットデバイスがFPUを持たない場合の発生例:

```
-----  
#include
```

```
void func(void) {  
    double result;
```

```
double x = -1.00000001; /* 発生条件(2) */  
double y = 4294967295ul; /* 発生条件(3) */
```

```
    result = pow(x, y);  
}
```

resultの値が-4.49579e+018ではなく+4.49579e+018になります。

2.4 回避策

回避策はありません。

2.5 恒久対策

今後のバージョンで改修予定です (次のリリース時期は未定です)。

[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。