

SuperH RISC engine C/C++コンパイラ Ver. 7.0.04.004 不具合内容

- 過去のお知らせ

本コンパイラの不具合内容を以下に示します。

以下不具合はチェックツールを使用することにより、プログラムに当該ケースが存在するか確認することができます。チェックツールは、ルネサス エレクトロニクス株式会社のホームページより入手できます。

http://tool-support.renesas.com/jpn/toolnews/shc/shcv7/dr_shcv7_1.html

1. R0 レジスタの不正破壊

【内容】

スタック渡しのパラメタがある時、R0 を不正に書き換える場合がある。

[例]

```
short func1(short a0, int *a1, int a2, short a3, short a4,
             short a5, short a6, int a7, int a8, int a9);
void func0(short a0, int *a1, int a2, short a3, short a4,
           short a5, short a6)
{
    :
    r1=func1(0, a1, 0, 0, 0, 0, 0, 0, 0);
    if((r1>0)&&(r1!=1)) {
        func1(a0, a1, 0, a3, a4, a5, a6, 0, 0, 0);
    }
    :
}
```

[出力コード]

```
MOV.L R0,@(32,R15) ; R0 を@(32,R15)に退避
MOV R8,R5
MOV #66,R0 ; R0 を破壊
MOV.W @(R0,R15),R3
MOV R9,R6
MOV #70,R0 ; R0 を破壊
MOV.W @(R0,R15),R1
MOV R0,R4 ; MOV.L @(32,R15),R4 を MOV R0,R4 に置換えコード不正
MOV.L R3,@(4,R15)
MOV.L R1,@(8,R15)
MOV.L R9,@(12,R15)
MOV.L R9,@(16,R15)
BSR _func1
MOV.L R9,@(20,R15)
:
```

【発生条件】

以下の条件をすべて満たす場合、発生することがあります。

- (1) optimize=1 を指定する。
- (2) 当該関数にスタック渡しのパラメタが存在する。

【回避方法】

以下のいずれかの方法で当該ケース回避することができます。

- (1) チェックツールによって当該ケースが見つかったプログラムを `optimize=0` でコンパイルする。
- (2) スタック渡しパラメータを関数先頭でローカル変数にコピーし、直後に `nop()` を入れる。関数内ではコピーしたローカル変数のみ参照する。

[例]

```
#include <machine.h> /* for nop() */

void func0(short a0, int *a1, int a2, short a3, short a4, short a5, short a6)
{
    short r1;

    /* 関数内処理の先頭でスタック渡しパラメータをローカル変数へコピー */
    short tmp4=a4, tmp5=a5, tmp6=a6;

    /* その直後に nop() を挿入 */
    nop();

    /* これ以降ではスタック渡しパラメータを参照せずコピーしたものを参照する */
    :
    r1=func1(0, a1, 0, 0, 0, 0, 0, 0, 0, 0);
    if((r1>0)&&(r1!=1)) {
        /* a4-a6 を tmp4-tmp6 へ変更 */
        func1(a0, a1, 0, a3, tmp4, tmp5, tmp6, 0, 0, 0);
    }
    :
}
```

2. BRA 命令飛び先不正

【内容】

無条件分岐を含むプログラムにおいて、分岐が BRA 命令で、かつ飛び先までの距離が 4094 バイトの時、飛び先が不正になるコードを生成する場合があります。

【発生条件】

以下の条件をすべて満たす場合、発生することがあります。

- (1) `code=machinecode` を指定している。または `code` オプションを指定していない。
- (2) BRA 命令から飛び先までの距離が 4094 バイトである。

【回避方法】

以下のいずれかの方法で当該ケースを回避することができます。

- (1) チェックツールによって当該ケースが見つかったプログラムを `code=asmcode` 指定してコンパイルする。
- (2) チェックツールによって当該ケースが見つかった関数に `nop()` を挿入することにより回避できる場合があります。

[例]

```
void func(int a) {
    if (a) {
```

```
        : /* 判定が偽の時、4094 バイト先へ飛ぶ BRA 命令になる */  
    }  
}  
  
↓  
  
#include <machine.h> /* for nop() */  
void func(int a) {  
    if (a) {  
        :  
        nop(); /* nop() を挿入する */  
    }  
}
```

 株式会社 日立超LSIシステムズ

(c) Hitachi ULSI Systems Co., Ltd. 1995,2014. All rights reserved.