

M3T-CC32R ご使用上のお願い

M32Rファミリ用CコンパイラパッケージM3T-CC32Rの使用上の注意事項を連絡します。

- キャストにより型変換した関数のポインタで関数呼び出しする場合の注意事項

1. 該当製品

M3T-CC32R V.1.00 Release 1 ~ V.4.30 Release 00

2. 内容

キャストにより型変換された関数へのポインタを使って関数を呼び出した場合、関数の仮引数と実引数の型が適合するにもかかわらず、次のエラーを発生することがあります。

```
error: type of argument does not match with prototype
```

3. 発生条件

次の条件すべてに該当する場合に発生します。

- (1) キャストにより、任意のポインタまたは整数を関数へのポインタに型変換し、同時に関数呼び出しを行っている。
- (2) 呼び出される関数の仮引数のうち少なくともひとつが配列型である。

発生例1: sample1.c

```
-----  
short array[3];
```

```
void foo1(void *ptr1)  
{  
    (*(void(*) (int, short arr[]))ptr1)(2,array);    /* 条件(1)(2) */  
}
```

発生例2: sample2.c

```
typedef struct AAA TYPE_A;
char *ptr2;

int foo2(TYPE_A array[][3])
{
    int ans;
    ans = (*(int(*) (TYPE_A arr[][3]))ptr2)(array); /* 条件(1)(2) */
    return ans;
}
-----
```

発生例3: sample3.c

```
typedef float TYPE_F[3];
TYPE_F array;

void foo3(void)
{
    (*(void(*) (TYPE_F))0x123400)(array); /* 条件(1)(2) */
}
-----
```

4. 回避策

キャストにより変換された関数へのポインタを使って関数を呼び出す場合はいったん別のポインタ変数にキャスト結果を格納して、この変数を使って関数呼び出しを行ってください。

sample1.c の回避例

```
short array[3];

void foo1(void *ptr1)
{
    void (*callptr1)(int, short arr[]) /* ポインタ変数callptr1を定義 */
    = (void (*)(int,short arr[]))ptr1;
    /* キャスト結果をcallptr1に格納 */
    (*callptr1)(2,array); /* callptr1を使って関数呼び出し */
}
-----
```

sample2.c の回避例

```
typedef struct AAA TYPE_A;
char *ptr2;

int foo2(TYPE_A array[][3])
{
    int ans;
    int (*callptr2)(TYPE_A arr[][3]); /* ポインタ変数callptr2を定義 */
    callptr2 = (int (*)(TYPE_A arr[][3]))ptr2;
        /* キャスト結果をcallptr2に格納 */
    ans = (*callptr2)(array);      /* callptr2を使って関数呼び出し */
    return ans;
}
```

sample3.c の回避例

```
typedef float TYPE_F[3];
void *ptr3;
TYPE_F array;

void foo3(void)
{
    void (*callptr3)(TYPE_F);      /* ポインタ変数callptr3を定義 */
    callptr3 = (void (*)(TYPE_F))0x123400;
        /* キャスト結果をcallptr3に格納 */
    (*callptr3)(array);          /* callptr3を使って関数呼び出し */
}
```

5. 恒久対策

本問題は、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。