

## M32Rファミリ用C/C++コンパイラパッケージ M3T-CC32R ご使用上のお願い

M32Rファミリ用C/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。

- ポインタと整数0を!=演算子で比較する場合の注意事項

### 1. 該当製品

M32Rファミリ用C/C++コンパイラパッケージ M3T-CC32R  
V.4.00 Release 1 ~ V.5.00 Release 00

### 2. 内容

以下の条件に該当する場合、NULLまたは0に動的に初期化されるポインタと整数の0を!=演算子で比較すると、その結果が常に真になります。

#### 2.1 発生条件

以下の条件をすべて満たす場合に問題が発生します。

- (1) 最適化オプション-O7,-O6,-O5 および-O4のいずれかを指定している。または、-Ospaceまたは-Otimeを単独で指定している。
- (2) ある関数内にNULLマクロまたは0に初期化されるポインタがある。
- (3) (2)と同じ関数内で、(2)のポインタと整数の0を!=演算子で比較し、比較結果を次の(a)または(b)いずれかで用いている。
  - (a) if文の条件
  - (b) 演算結果が参照されている関係演算子(?:)の条件

## 2.2 発生例

### 例1

```
-----  
-----  
void func1(void)  
{  
    void (*p_func1)(void);  
    p_func1 = 0;          /* 発生条件(2) */  
  
    if (p_func1 != 0) {   /* 発生条件(3)(a) */  
        (*p_func1)();  
    }  
}
```

### 例2

```
-----  
-----  
#include <stddef.h>  
short *p_short;  
void func2(void)  
{  
    p_short = NULL;      /* 発生条件(2) */  
  
    if (0 != p_short) {  /* 発生条件(3)(a) */  
        *p_short = 100;  
    }  
}
```

### 例3

```
-----  
-----  
#include <stddef.h>  
int func3(void)  
{  
    int short *p_int = NULL; /* 発生条件(2) */  
  
    return (p_int!=0 ? 1 : 2); /* 発生条件(3)(b) */  
}
```

-----  
-----

### 3. 回避策

次のいずれかの方法で回避できます。

- (1) != 演算子のオペランド0をキャスト((void\*))を付加するか、NULLマクロに変更する。

例1の回避例

```
-----  
-----  
void func1(void)  
{  
    void (*p_func1)(void);  
    p_func1 = 0;  
  
    if (p_func1 != (void*)0) { /* != 0 を != (void*)0 に  
変更 */  
        (*p_func1)();  
    }  
}
```

例2の回避例

```
-----  
-----  
#include <stddef.h>  
short *p_short;  
void func2(void)  
{  
    p_short = NULL;  
  
    if (NULL != p_short) { /* 0 != を NULL!= に変更  
*/  
        *p_short = 100;  
    }  
}
```

-----  
-----

### 例3の回避例

```
-----  
-----  
#include <stddef.h>  
int func3(void)  
{  
    int short *p_int = NULL;  
  
    return (p_int !=(void*)0 ? 1 : 2); /* != 0 を  
!=(void*)0 に変更 */  
}  
-----  
-----
```

- (2) 次のいずれかの方法でレベル4の最適化を抑止する。
- (a) -O7, -O6, -O5, および-O4のいずれかを指定している場合は-O3, -O2, -O1, および-O0のいずれかに変更する。
  - (b) -Ospaceまたは-Otimeを指定する場合は、同時に-O3, -O2, -O1, および-O0 のいずれかを使用する。

## 4. 恒久対策

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。