

32bit×32bit→64bit演算の注意

IAR社のEWARMでは64bit変数(long long)がサポートされているので32bit変数同士のかけ算で64bit結果を得る時の留意点についての備忘録。(他のコンパイラでは未確認)

ソース例 1 でアセンブラ出力を確認

下記記述では、「MULS」命令で32bit演算命令が使われてしまっている。

[dat64.c](#)

```
int64_t dmuls(int32_t dat1, int32_t dat2)
{
    return (int64_t)(dat1 * dat2);
}
```

↓ (アセンブラ出力リスト)

```
49      int64_t dmuls(int32_t dat1, int32_t dat2)
50      {
51          return (int64_t)(dat1 * dat2);
           dmuls: (+1)
           0x0    0x4348      MULS    R0,R1,R0
           0x2    0x17C1      ASRS    R1,R0,#+31
           0x4    0x4770      BX      LR                ;; return
52      }
```

ソース例 2 でアセンブラ出力を確認

下記記述では、「SMULL」命令で期待通り64bit演算命令が使われた。

[dat64.c](#)

```
int64_t dmuls(int32_t dat1, int32_t dat2)
{
    return (int64_t)dat1 * (int64_t)dat2;
}
```

↓ (アセンブラ出力リスト)

```
49      int64_t dmuls(int32_t dat1, int32_t dat2)
50      {
51          return (int64_t)dat1 * (int64_t)dat2;
```

```
dmuls: (+1)
    0x0    0xFB81 0x0100    SMULL    R0,R1,R1,R0
    0x4    0x4770    BX        LR        ;; return
52      }
```

64bit結果を得たい時には、上記のように非演算側をキャストする必要があるので注意。

From:
<https://www.yokoweb.net/dokuwiki/> - 新石器Wiki

Permanent link:
<https://www.yokoweb.net/dokuwiki/programing/c-language/64bit-multiply-instruction>

Last update: 2026/07/28 13:16

