# Newlib线程不安全问题

环境：GCC（典型环境：RT-Studio）

版本：截至目前的RTT任何版本

newlib的strtod、sprintf、sscanf、snprintf、vsnprintf这些函数都是非线程安全的，这些函数的行为和keil\IAR以及我们的认知并不一样，这些函数在使用%f %lf来输出浮点时，会在申请一块3.5K左右的内存空间，该内存空间只申请一次，且不释放。作为全局内存使用。如果有多个线程同时调用sprintf来字符串化浮点数据，会有几率或较大几率导致系统死机（尤其是多个线程高密度的调用sprintf来输出浮点数据）。如果仅用来输出字符串或整形数据，则不会出该问题，即该问题仅局限于浮点数这一块。

**即：**sprintf、sscanf、snprintf、vsnprintf、sscanf这些函数在不处理浮点数相关的时候还是线程安全的，一旦涉及到浮点数%f %lf就会申请一块公共的内存空间(3.5KB)导致线程的不安全；而strtod以及printf是彻头彻底的线程不安全（printf线程不安全是正常的，printf在哪个编译链下都是线程不安全的；而strtod sprintf等函数理论上都应该是线程安全的 <https://manpages.courier-mta.org/htmlman3/strtod.3.html>）。

由于RT-Thread早期大部分工程师都是基于IAR或者Keil来开发，自从studio IDE开发出来之后有让更多工程师使用GCC的机会，使得这个问题暴露了出来。

这种问题死机有一个很明显的特征就是用CmBacktrace来追踪时 会有一个叫\_Balloc 的函数，如果出现这个函数了就是因为newlib浮点线程不安全导致的。

* newlib的这个问题也被FreeRTOS Arduino等使用newlib的社区所报告。
* zephyr也遇到了相同的问题，并给出的建议issue: <https://github.com/zephyrproject-rtos/zephyr/issues/21519> 。以及尝试解决的 <https://github.com/zephyrproject-rtos/zephyr/pull/21518> 但是很遗憾，该PR被关闭，截止目前Zephyr仍然没有解决这个问题。
  + 被关闭的PR一定要重点留意一下，Zephyr关闭担心的理由很有可能也是我们需要担心的理由。
  + zephyr宣布在<https://github.com/zephyrproject-rtos/zephyr/pull/36201> 中解决了这个问题。并声称reentrancy support现在还不需要，因此不需要实现。[https://github.com/zephyrproject-rtos/zephyr/issues/21519#issuecomment-892449637](https://github.com/zephyrproject-rtos/zephyr/issues/21519" \l "issuecomment-892449637) 。但是经过确认，zephyr并**没有**将本文所提到的sprintf涉及到浮点数死机的问题解决，实际上这些mutex函数在newlib很少被用到。经过测试，在调用sprintf时，也没有任何mutex函数被调用。
    - 上一点中的zephyr解决方案也有FreeRTOS版本的实现 <https://gist.github.com/thomask77/3a2d54a482c294beec5d87730e163bdd>
    - <https://github.com/mirror/newlib-cygwin/blob/436e47584ae7aecf0ec22a98384a3348d244471f/newlib/libc/misc/lock.c> newlib官方的描述
    - 对\_\_retarget\_lock的讨论（newlib官方的回答）：<https://sourceware.org/legacy-ml/newlib/2018/msg00827.html>
    - 综上，zephyr实际解决的是FILE文件结构体的多线程共享问题，和sprintf没有一毛钱关系。

**参考网站：**

* <https://nadler.com/embedded/newlibAndFreeRTOS.html>
* <https://www.freertos.org/FreeRTOS_Support_Forum_Archive/April_2018/freertos_How_to_make_printf_sprintf_strtod_thread_safe_44d2ceb8j.html>
* [When to enable configUSE\_NEWLIB\_REENTRANT macro? - FreeRTOS](https://www.freertos.org/FreeRTOS_Support_Forum_Archive/July_2019/freertos_When_to_enable_configUSE_NEWLIB_REENTRANT_macro_7b689bc63bj.html)
* <http://nickhung0807.blogspot.com/2014/06/newlib-reentrancy.html>

\_impure\_ptr全局性指针

[\_impure\_ptr not setup when entering first task - FreeRTOS](https://www.freertos.org/FreeRTOS_Support_Forum_Archive/October_2013/freertos_impure_ptr_not_setup_when_entering_first_task_b771391dj.html)

[Embedding with GNU: Newlib - Embedded.com](https://www.embedded.com/embedding-with-gnu-newlib/)

**常规解决方案**

需要定义\_reent结构体，该结构体含有96个字节，如下图所示，每个线程以及ISR环境都需要定义，会导致ram的急剧上升。very expensive！实际上是不止96个字节的，因为这个结构体里边含有一些指针，可能内部还会基于这些指针继续分配内存。

* 在FreeRTOS仓库里搜索configUSE\_NEWLIB\_REENTRANT和\_reclaim\_reent关键词

图形用户界面

中度可信度描述已自动生成

该\_reent结构体在FreeRTOS中的定义

文本

描述已自动生成

**\_REENT\_ONLY 宏定义**

[J. Johnston - Re: \_REENT\_ONLY question (sourceware.org)](https://sourceware.org/legacy-ml/newlib/2003/msg00697.html)

此外math.h缺少一些函数如pow，貌似也和这事有联系

[msp430 | BIG problems with math.h (embeddedrelated.com)](https://www.embeddedrelated.com/showthread/msp430/16987-1.php)

**在Keil下也可能需要申请类似的96个自己的私有变量空间**

[Libraries and Floating Point Support Guide: Use of the \_\_user\_libspace static data area by the C libraries (keil.com)](https://www.keil.com/support/man/docs/armlib/armlib_chr1358938914047.htm)

<https://www.keil.com/support/man/docs/armlib/armlib_chr1359122864707.htm>

<https://www.keil.com/support/man/docs/armlib/armlib_chr1358938915795.htm>

<https://www.keil.com/support/man/docs/armlib/armlib_chr1358938914921.htm>

https://www.keil.com/support/man/docs/armlib/armlib\_chr1358938915514.htm

[ARM Developer Suite Compilers and Libraries Guide](https://developer.arm.com/documentation/dui0067/d/the-c-and-c---libraries/tailoring-storage-management/creating-your-own-storage-management-system)

经过实际测试，keil并不存在这种问题，sprintf snprintf vsnprintf都是线程安全的！

因此keil的96字节可以不用实现。

Newlib的\_REENT\_ONLY作用

<https://sourceware.org/legacy-ml/newlib/2003/msg00697.html>

这里边解释了\_REENT\_ONLY宏的作用，主要是强制用户必须使用可重入版本的库函数，以避免线程不安全的问题。

# Newlib下memcpy/memset函数可能导致的死机问题

在newlib下面的memcpy 以及memset等函数是不考虑字节对齐问题的，这可能会导致指针地址为非字节对齐时直接死机。

详见：<https://github.com/lvgl/lvgl/issues/2790#issuecomment-980479598>

# Newlib下malloc等函数

Newlib下 malloc这块也有一堆问题，但是幸运的是rt-thread直接采用拦截malloc等函数自己进行实现，因此把这些问题都直接避免掉了。