关于程序为什么会偶发进入HardFault的调查

【事实】

1. 现象

程序偶发HardFault
频率：（空跑状态）有时1个小时1次，有时3个小时1次

2. 目标硬件环境

MCU：GD32F107VCT6
SystemCoreClock：108000000Hz
RTC：Use
SPI-Flash ：Use
Eth(Phy:LAN8720A)：Use
USART1,USART2(DMA)：Use
Timer2-Timer7：Enable（按需分配给任务,空跑状态下不使用）

3. 目标软件环境

RTOS：FreeRTOS V9.0.0
TCP/IP Stack：LWIP 1.4.1
Running User Task number：10

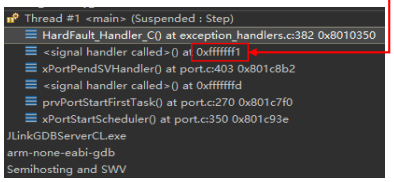
4. 调试环境

调试器：Jlink SWD
开发环境：eclipse+gnuarmeclipse plugin

4. 进入HardFault时的Dump

Stack frame: **CallStack**

R0 = 0x20018020
R1 = 0x0
R2 = 0x0
R3 = 0x2000e0a8
R12 = 0xa5a5a5a5
LR = 0xfffffffd
PC = 0x801c8b2
PSR = 0x100800e



FSR/FAR:

CFSR = 0x00008200
HFSR = 0x40000000
AFSR = 0x0
BFAR = 0x20018000

Misc

LR/EXC_RETURN = 0xfffffff1

【分析】

从进入HardFault时的调用栈来看，连续有两次的<signal handler called> at 0xfffffff1(or 0xfffffffd)
=> 从地址来看，是否是因为发生了锁定？(ARM Cortex-M3 与Cortex-M4权威指南 12.7章 锁定)
=> 书中介绍只有在硬件错误或NMI处理期间产生的错误才会引起锁定
=> 如何判断是否发生了锁定？发生锁定的原因如何排查还不太清楚

从CFSR寄存器来看，值=0x00008200

(高16略)	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	BFARVAILD						PRECISERR									
	↓						↓									
	BFAR合法						精确的数据访问冲突									

=> 根据Cortex-M3与Cortex-M4权威指南(12.4.3 总线错误信息)中介绍可以通过BFSR寄存器读取发生数据错误的地址。

从BFSR寄存器来看，值=0x20018000
=> 从下面URL(iar japan)中的介绍(例3)，也能肯定当bit9被置1的时候出错的地址是会保存在BFSR寄存器中
http://www.iarsys.co.jp/customer/Faqs/faqListView/10810531
=> 如何从该内存判断到底是那一个数组?变量?任务栈?出的问题呢？

从PC地址分析
=> 先分析Banded是哪个栈指针(通过LR寄存器的值的bit2判断的(bit2=0->MSP，bit2=1->PSP))
=> LR寄存器的值=0xfffffffd,bit2=0，是MSP，但是不是一个有效地址，无法判断。

额外的信息：
1）DFSR 调试错误状态寄存器 = 3，即BKPT位=1（调试事件由断点引起）
=> 就是停在HardFault的断点处，所以该信息无参考意义。

2）HFSR 硬件错误状态寄存器 = 0x40000000，即FORCED(bit30位)=1
（表明硬件错误由于总线错误,存储器管理错误或者使用错误引发）
=> ARM Cortex-M3 与Cortex-M4权威指南 12.5章介绍：是由一个可配置错误引起，应该检查CFSR
=> 最上面的分析已经检查过了。

疑问点
1）因为调用栈都是FreeRTOS的API，是否是由于FreeRTOSConfig.h中的配置有问题呢？
=> 特别是configKERNEL_INTERRUPT_PRIORITY，configMAX_SYSCALL_INTERRUPT_PRIORITY等参数
=> 见Sheet(FreeRTOSConfig)

2）GD32F107的主频从72Mhz抬高到了108MHz，是否是因为频率太快？与RTOS的配置或者其他外设的时钟不匹配造成的？
=> 恢复到72Mhz试一下。

【总结&ACTION】

还未判明原因，暂不记入

先降低主频再试一次。

```
extern uint32_t SystemCoreClock;

#define configUSE_PREEMPTION		1
#define configUSE_TIME_SLICING		1
#define configUSE_IDLE_HOOK			0
#define configUSE_TICK_HOOK			0
#define configCPU_CLOCK_HZ			( SystemCoreClock )
#define configTICK_RATE_HZ			( ( TickType_t ) 1000 )
#define configMAX_PRIORITIES		( 15 )
#define configMINIMAL_STACK_SIZE	( ( unsigned short ) 128 )
#define configTOTAL_HEAP_SIZE		( ( size_t ) ( 40 * 1024 ) )
#define configMAX_TASK_NAME_LEN		( 16 )
#define configUSE_TRACE_FACILITY	1
#define configUSE_STATS_FORMATTING_FUNCTIONS 1
#define configUSE_16_BIT_TICKS		0
#define configIDLE_SHOULD_YIELD		0

#define configUSE_COUNTING_SEMAPHORES	1
#define configCHECK_FOR_STACK_OVERFLOW	2

/* Co-routine definitions. */
#define configUSE_CO_ROUTINES		0
#define configMAX_CO_ROUTINE_PRIORITIES ( 2 )

/* Set the following definitions to 1 to include the API function, or zero
 to exclude the API function. */

#define INCLUDE_vTaskPrioritySet		1
#define INCLUDE_uxTaskPriorityGet		1
#define INCLUDE_vTaskDelete			1
#define INCLUDE_vTaskCleanUpResources	0
#define INCLUDE_vTaskSuspend			1
#define INCLUDE_vTaskDelayUntil		1
#define INCLUDE_vTaskDelay			1

#define INCLUDE_xTaskGetCurrentTaskHandle	1
#define INCLUDE_uxTaskGetStackHighWaterMark	1

#define configUSE_MUTEXES	1

/* This is the raw value as per the Cortex-M3 NVIC.  Values can be 255
 (lowest) to 0 (1?) (highest). */
#define configKERNEL_INTERRUPT_PRIORITY		255
/* !!!! configMAX_SYSCALL_INTERRUPT_PRIORITY must not be set to zero !!!!
 See http://www.FreeRTOS.org/RTOS-Cortex-M3-M4.html. */
#define configMAX_SYSCALL_INTERRUPT_PRIORITY	191 /* equivalent to 0xb0, or priority 11. */

/* This is the value being used as per the ST library which permits 16
 priority values, 0 to 15.  This must correspond to the
 configKERNEL_INTERRUPT_PRIORITY setting.  Here 15 corresponds to the lowest
 NVIC value of 255. */
#define configLIBRARY_KERNEL_INTERRUPT_PRIORITY 15

#endif /* FREERTOS_CONFIG_H */
```