

mma内存使用debug参考

mma alloc fail如何处理

1. 把sys的debug_level提高到MI_DBG_KMSG级别，当alloc mma fail时会自动打印mma的使用情况：

```
/ # echo 3 > /proc/mi_modules/mi_sys/debug_level
/ #
```

```
typedef enum
{
    ... MI_DBG_NONE = 0,
    ... MI_DBG_ERR,
    ... MI_DBG_WRN,
    ... MI_DBG_KMSG,
    ... MI_DBG_API,
    ... MI_DBG_INFO,
    ... MI_DBG_DEBUG,
    ... MI_DBG_TRACE,
    ... MI_DBG_ALL
}MI_DBG_LEVEL_e;
```

2. 如果确实是chunk_mgr_avail不够了，那就需要在config中增大sz的值：

```
MHAL = i2m
MERGE_BOOT = TRUE
BOOTLOGO_FILE = sigmastar1024_600.jpg
BOOTLOGO_ADDR = E_LX_FB
DISP_OUT_NAME = SAT070CP50
EXBOOTARGS =
KERNEL_BOOT_ENV = LX_MEM=$(KERNEL_MEMLN) mma_heap=mma_heap_name0,miu=0,sz=0x3800000 mma_memblock_remove=1
TOOLCHAIN_REL = arm-linux-gnueabihf-
```

3. 如果是chunk_mgr_avail还够，但是没有连续的一整块buf足够alloc(内存碎片)，可以打开MMU,优化内存碎片问题，提高mma使用率
打开MMU的方法很简单，只需要在config中增加MMU的设置即可：

```
DISP_OUT_NAME = SAT070CP50
EXBOOTARGS =
KERNEL_BOOT_ENV = LX_MEM=$(KERNEL_MEMLN) mma_heap=MMU_MMA,miu=0,sz=0x3800000 mma_heap=mma_heap_name0,miu=0,sz=1000000 mma_memblock_remove=1
TOOLCHAIN_REL = arm-linux-gnueabihf-
```

其中sz为对应MMU的大小，打开MMU后，大部分sdk的内存会到MMU里面来alloc，所以可以把MMU_MMA的大小设定成接近mma_heap_name0的大小，然后使用cat /proc/mi_modules/mi_sys_mma/MMU_MMA，查看MMU的使用情况来决定需要分配多大的内存给MMU的大小来确认需要分配的MMU大小