

## 说明：

- oom (out of memry) 机制是操作系统用来保证操作系统自身正常运行的一个机制
- 它是通过kill 掉一些进程来保证系统剩余内存足够系统正常运行

## 限制：

/proc/sys/vm/min\_free\_kbytes

1. 通过设置 min\_free\_kbytes 来 限制Linux VM最低保留多少空闲内存 (Kbytes) ；
2. 当可用内存低于这个参数时，系统开始kill 进程，以释放内存，直到可用内存大于这个值

### 该值不能设置的过小，当该值设置的过小时

1. 内存使用过量时，没有及时kill 掉程序，导致 频繁换页 降低页命中率，系统运行卡顿，
2. 测试发现：会增加系统死机的概率，当内存太少，内核可能会直接死机，串口无任何反应，

### 该值也不能设置过大

1. 设置过大，会导致应用层能使用的内存减少，

## 问题查找：

当内核报oom时，如何查找问题

- 1: 是一个内存泄漏导致的OOM，
- 2: 还是应用本身设计不当，导致一次需要加载的内存过多，导致的OOM？

在嵌入式开发中，小内存设备可能会出现第二种情况；

1. 程序能够正常跑起来 一段时间后内核报 oom，看内存是不是一直在增长，如果在使用的过程中内存一直在增长，则很有可能是内存泄漏导致的。
2. 程序不能正常跑起来，可能是情况1 也有可能是情况2

## 临时处理办法:

1: 设置不能被kill掉的 进程 oom\_score\_adj

设置 `echo -1000 /proc/<pid>/oom_score_adj`

设置 oom\_adj 在 新内核上 已经过时了

/\*

\* /proc/<pid>/oom\_score\_adj set to OOM\_SCORE\_ADJ\_MIN disables oom killing for

\* pid.

\*/

#define OOM\_SCORE\_ADJ\_MIN (-1000)

## 测试方法:

手动触发 oom, 测试

`echo f > /proc/sysrq-trigger`