1.**/etc/sysctl.conf**  
前面启动Redis时，看到如下警告：

引用

[1958] 13 Aug 16:18:24 # WARNING overcommit\_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit\_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit\_memory=1' for this to take effect.

需要修改**/etc/sysctl.conf**文件：

Shell代码  [收藏代码](javascript:void())

1. vim /etc/sysctl.conf

末尾追加**vm.overcommit\_memory = 1**  
然后执行**sysctl vm.overcommit\_memory=1**，使之生效：

Shell代码  [收藏代码](javascript:void())

1. # sysctl vm.overcommit\_memory=1
2. vm.overcommit\_memory = 1

2.**/proc/sys/vm/overcommit\_memory**  
为了调整内存分配策略，需要配置**/proc/sys/vm/overcommit\_memory**

* 0， 表示内核将检查是否有足够的可用内存供应用进程使用；如果有足够的可用内存，内存申请允许；否则，内存申请失败，并把错误返回给应用进程。
* 1， 表示内核允许分配所有的物理内存，而不管当前的内存状态如何。
* 2， 表示内核允许分配超过所有物理内存和交换空间总和的内存

默认为0，如果内存情况比较紧张的话，设为1：

Shell代码  [收藏代码](javascript:void())

1. echo 1 > /proc/sys/vm/overcommit\_memory

3.**redis.conf**  
前面启动Redis后，总是在命令行里不断跳着各种日志，很麻烦。即便通过“&”，领其后台运行，也无济于事。这就需要修改**redis.conf**，以Daemo模式运行！  
**redis.conf**参数：

* daemonize：是否以后台daemon方式运行
* pidfile：pid文件位置
* port：监听的端口号
* timeout：请求超时时间
* loglevel：log信息级别
* logfile：log文件位置
* databases：开启数据库的数量
* save \* \*：保存快照的频率，第一个\*表示多长时间（秒级），第三个\*表示执行多少次写操作。在一定时间内执行一定数量的写操作时，自动保存快照。可设置多个条件。
* rdbcompression：是否使用压缩
* dbfilename：数据快照文件名（只是文件名，不包括目录）
* dir：数据快照的保存目录（这个是目录）
* appendonly：是否开启appendonlylog，开启的话每次写操作会记一条log，这会提高数据抗风险能力，但影响效率。
* appendfsync：appendonlylog如何同步到磁盘（三个选项，分别是每次写都强制调用fsync、每秒启用一次fsync、不调用fsync等待系统自己同步）
* slaveof <masterip> <masterport> ：主从配置，在redis-slave上配置master的ip port，即可。

例如，我们可以修改为如下方式：

引用

daemonize yes #守护进程模式  
save 60 1000 #当时间间隔超过60秒，或存储超过1000条记录时，进行持久化。  
maxmemory 256mb #分配256MB内存

PS：切记，一定要设定**maxmemmory**，且配置大小要小于物理内存，留有足够的内存供系统使用。

公司一同学的Redis，某期间数据暴涨，导致内存吃紧，SWAP加剧，直接宕机。就是因为没有设置**maxmemmory**。