

Android 系统的分区

- 1.modem分区，主要是手机短信的通讯模块；
- 2.bootloader分区，主要用于线刷；
- 3.recovery分区，只有用于普通刷机；
- 4.boot分区，用于启动；
- 5.system分区，系统分区；
- 6.data分区，用户分区；
- 7.cache分区：存储使用过程中的缓存数据
- 8.vendor：包含有厂商私有的可执行程序、库、系统服务和app等。可以将此分区看做是system分区的补充，厂商定制ROM的一些功能都可以放在此分区。

双清就是指对data分区和cache分区进行清理。

modem分区
bootloader分区
boot分区
recovery分区
system分区
data分区
cache分区
vendor分区

内容来源: csdn.net
作者昵称: 菜鸟xiaowang
原文链接: https://blog.csdn.net/xiaowang_lj/article/details/125430257
作者主页: https://blog.csdn.net/xiaowang_lj

boot分区：由 kernel 和 ramdisk 构成，内核就在这里

system分区：这个分区包含了整个操作系统除 kernel 和 ramdisk 之外的其它部分，包括 Android UI 和设备上预装的 APP。**删除这个分区只会同步将 Android 系统从设备上卸载掉**，而不会导致设备不能重启，因此你依然可以顺利让设备进入到 recovery 或 bootloader 模式来为它重新刷入一个新的 ROM。

recovery 分区：相当于一个备用的 boot 分区，利用它我们可以使设备启动到恢复模式窗口来进行一些高级的恢复和维护操作。

data分区：也被称作 userdata，是用户数据被存储的地方——也就是说你的联系人、短信、配置文件和你自己安装的 APP 都位于这里。**删除这个分区相当于将你的手机恢复出厂设置**，你的设备会变得和你第一次开机或最后一次刷入官方/自定义 ROM 后一样。你在设备上点击清除用户数据或恢复出厂设置按钮后，实际上清除掉的就是这个分区。

cache分区：存放的是 Android 系统中经常被访问的数据和 APP 的部件。删除这个分区不会对你的个人数据造成影响，这个分区上的数据会随着你继续使用设备被重新收集并存储在这里。

misc分区：包含了各种以开关量形式保存的系统配置，包括 CID（Carrier or Region ID）、USB配置和某些硬件配置等。这是个很重要的分区，如果它遭到了破坏或有数据丢失，那么设备的一些功能可能无法正常使用

Android系统的启动

- 启动：
- 概述：Loader > Kernel > Native > Framework > Application
- 细分：BootRom > Bootloader > Kernel > Init > Zygote > SystemServer > Launcher
- Loader层主要包括Boot Rom和Boot Loader
- Kernel层主要是Android内核层
- Native层主要是包括init进程以及其fork出来的用户空间的守护进程、HAL层、开机动画等
- Framework层主要是AMS和PMS等Service的初始化
- Application层主要指SystemUI、Launcher的启动

bootloader分区分成两个部分，分别叫做primary bootloader和secondary stage bootloader。Primary bootloader主要执行硬件检测，确保硬件能正常工作后将secondary stage bootloader拷贝到内存(RAM)开始执行。Secondary stage bootloader会进行一些硬件初始化工作，获取内存大小信息等，然后根据用户的按键进入到某种启动模式

fastboot模式：fastboot是android定义的一种简单的刷机协议，用户可以通过fastboot命令行工具来进行刷机。比如说fastboot flash boot boot.img这个命令就是把boot.img的内容刷写到boot分区中

recovery模式：recovery是android定义的一个标准刷机协议。当进入recovery模式时，secondary stage bootloader从recovery分区开始启动，recovery分区实际上是一个简单的Linux系统，当内核启动完毕后，开始执行第一个程序init(init程序是Linux系统所有程序的老祖宗)。init会启动一个叫做recovery的程序（recovery模式的名称也由此而来）。通过recovery程序，用户可以执行清除数据，安装刷机包等操作

手机除了普通的CPU芯片以外，还有MODEM处理器芯片。该芯片的功能就是实现手机必需的通信功能，大家通常所说的刷RADIO就是刷写modem分区。

内容来源: csdn.net
作者昵称: 菜鸟xiaowang
原文链接: https://blog.csdn.net/xiaowang_lj/article/details/125430257
作者主页: https://blog.csdn.net/xiaowang_lj

非A/B升级

- Android系统收到服务端下发的OTA推送，将OTA包下载至cache分区。
- OTA包下载完成后，将向misc分区写入指令，表明下次启动时进入recovery模式并使用该OTA包进行升级。
- 重启手机。
- 重启后最先进入bootloader，bootloader会先判断按键组合、电源寄存器等，随后会读取misc分区的内容并解析。由于步骤2中已经向misc分区写入了指令，此处bootloader读取指令后会引导启动recovery系统。
- 进入recovery，读取cache分区中的OTA包，并解析其中的升级脚本，按照其指令对系统各个分区进行升级。如果recovery自身也需要升级，会在此过程中向system中写入recovery-from-boot.p文件，这是一个recovery升级所需要的patch。
- recovery会清除misc分区。
- 重启手机。
- 重启后最先进入bootloader，判断按键组合、电源寄存器、misc分区内容等，默认情况会启动Android系统，此时已经是OTA升级后的新版本系统。
- 新版本Android系统启动后，会检查是否存在recovery-from-boot.p文件，如果存在，则会对recovery进行升级。

A/B升级

在Android O之后，Google引入了一种新的分区结构，称为A/B分区，与之对应，传统分区结构被称为non-A/B分区。

A/B分区结构，顾名思义，将系统分区分成了A和B两个槽(slot)，手机启动时会选择A槽或者B槽启动，运行过程中仅使用当前槽位的分区。一旦当前运行的槽出现问题，系统仍可以选择另一个槽进行启动，从而保证系统良好的可用性。

采用A/B分区结构，能够实现无缝升级。例如用户正在运行A槽，此时收到OTA推送，则系统会在后台一边下载OTA数据，一边同时对B槽进行升级。当B槽系统升级完成，用户会收到重启提示，此时重启手机将自动切换到B槽的新版本系统。在此过程中，仅重启操作是会被用户感知的，这个重启与普通重启的耗时没有什么区别。

如果OTA失败，也仅仅是待升级的槽出现问题，可以重新尝试OTA，并不会影响用户当前运行的系统。

由于A/B分区结构可以实现一边从服务端获取OTA数据，一边直接写入待升级的槽，不需要临时存储OTA包的空间，因此不再需要在cache或userdata分区预留足够空间。

内容来源: csdn.net

作者昵称: 慕鸟xiaowang_

原文链接: https://blog.csdn.net/xiaowang_lj/article/details/125430257

作者主页: https://blog.csdn.net/xiaowang_lj