[android系统分区](http://blog.csdn.net/zxh821112/article/details/50427759)

2015-12-29 16:51 113人阅读 [评论](http://blog.csdn.net/zxh821112/article/details/50427759#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/zxh821112/article/details/50427759#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

Android（3） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

Android系统开发时，经常会遇到添加或者调整系统分区大小的问题，下面以mstar的一款产品为例进行分析：

      （1）mount指令可以查看到板子中挂在的分区，主要关注ext4类型分区，例如tvservice，tvconfig，tvconfig，tvcustomer

**[plain]** [view plain](http://blog.csdn.net/zxh821112/article/details/50427759) [copy](http://blog.csdn.net/zxh821112/article/details/50427759)

1. 130|shell@arbutus:/ # mount
2. rootfs / rootfs ro,relatime 0 0
3. tmpfs /dev tmpfs rw,nosuid,relatime,mode=755 0 0
4. devpts /dev/pts devpts rw,relatime,mode=600 0 0
5. proc /proc proc rw,relatime 0 0
6. sysfs /sys sysfs rw,relatime 0 0
7. debugfs /sys/kernel/debug debugfs rw,relatime 0 0
8. none /acct cgroup rw,relatime,cpuacct 0 0
9. none /sys/fs/cgroup tmpfs rw,relatime,mode=750,gid=1000 0 0
10. tmpfs /mnt/asec tmpfs rw,relatime,mode=755,gid=1000 0 0
11. tmpfs /mnt/obb tmpfs rw,relatime,mode=755,gid=1000 0 0
12. none /dev/cpuctl cgroup rw,relatime,cpu 0 0
13. tmpfs /mnt/usb tmpfs rw,relatime,mode=755,gid=1000 0 0
14. tmpfs /mnt/iso tmpfs rw,relatime,mode=755,gid=1000 0 0
15. tmpfs /mnt/samba tmpfs rw,relatime,mode=755,gid=1000 0 0
16. tmpfs /var tmpfs rw,relatime,mode=775,gid=1000 0 0
17. /dev/block/platform/mstar\_mci.0/by-name/system /system ext4 ro,relatime,data=ordered 0 0
18. /dev/block/platform/mstar\_mci.0/by-name/cache /cache ext4 rw,nosuid,nodev,noatime,data=ordered 0 0
19. /dev/block/platform/mstar\_mci.0/by-name/userdata /data ext4 rw,nosuid,nodev,noatime,data=ordered 0 0
20. /dev/block/platform/mstar\_mci.0/by-name/tvservice /tvservice ext4 ro,relatime,data=ordered 0 0
21. /dev/block/platform/mstar\_mci.0/by-name/tvconfig /tvconfig ext4 ro,relatime,data=ordered 0 0
22. /dev/block/platform/mstar\_mci.0/by-name/tvdatabase /tvdatabase ext4 rw,nosuid,nodev,noatime,data=ordered 0 0
23. /dev/block/platform/mstar\_mci.0/by-name/tvcustomer /tvcustomer ext4 rw,nosuid,nodev,noatime,data=ordered 0 0
24. /dev/fuse /mnt/sdcard fuse rw,nosuid,nodev,noexec,relatime,user\_id=1023,group\_id=1023,default\_permissions,allow\_other 0 0

      （2）查看分区文件fstab.monet (device/mstar/arbutus/目录下)，截取部分信息如下：

**[plain]** [view plain](http://blog.csdn.net/zxh821112/article/details/50427759) [copy](http://blog.csdn.net/zxh821112/article/details/50427759)

1. /dev/block/platform/mstar\_mci.0/by-name/system          /system             ext4      ro                                                    wait
2. /dev/block/platform/mstar\_mci.0/by-name/cache           /cache              ext4      noatime,nosuid,nodev                                  wait,block\_validity,nodiscard,data=ordered,journal\_checksum
3. /dev/block/platform/mstar\_mci.0/by-name/userdata        /data               ext4      noatime,nosuid,nodev                                  wait,block\_validity,nodiscard,data=ordered,journal\_checksum
4. /dev/block/platform/mstar\_mci.0/by-name/tvservice       /tvservice          ext4      ro                                                    wait
5. /dev/block/platform/mstar\_mci.0/by-name/tvconfig        /tvconfig           ext4      ro                                                    wait
6. /dev/block/platform/mstar\_mci.0/by-name/tvdatabase      /tvdatabase         ext4      noatime,nosuid,nodev                                  wait,block\_validity,nodiscard,data=ordered,journal\_checksum
7. /dev/block/platform/mstar\_mci.0/by-name/tvcustomer      /tvcustomer         ext4      noatime,nosuid,nodev

      android平台不同厂商可能名称不一样，但是都是fstab.xxx的形式

      （3）指定各个分区的大小，BoardConfig.mk (device/mstar/arbutus)文件指定各自大小可以根据需要进行修改

**[plain]** [view plain](http://blog.csdn.net/zxh821112/article/details/50427759) [copy](http://blog.csdn.net/zxh821112/article/details/50427759)

1. # mmc part
2. # :  type name      (size)
3. #1.  misc           (512K)
4. #2.  recovery       (19M)
5. #3.  boot           (15M)
6. #4.  tee            (6M)
7. #5.  rtpm           (256K)
8. #6.  dtb            (1M)
9. #7.  system         (800M)
10. #8.  userdata       (900M)
11. #9.  cache          (1G)
12. #10. tvservice      (160M)
13. #11. tvconfig       (10M)
14. #12. tvdatabase     (8M)
15. #13. tvcustomer     (16M)
16. BOARD\_RECOVERYIMAGE\_PARTITION\_SIZE   := 0x01300000
17. BOARD\_BOOTIMAGE\_PARTITION\_SIZE       := 0x00F00000
18. BOARD\_TEEIMAGE\_PARTITION\_SIZE        := 0x00600000
19. BOARD\_RTPMIMAGE\_PARTITION\_SIZE       := 0x00040000
20. BOARD\_DTBIMAGE\_PARTITION\_SIZE        := 0x00100000
21. BOARD\_SYSTEMIMAGE\_PARTITION\_SIZE     := 0x33C00000
22. BOARD\_USERDATAIMAGE\_PARTITION\_SIZE   := 0x38000000
23. BOARD\_CACHEIMAGE\_PARTITION\_SIZE      := 0x40000000
24. BOARD\_TVSERVICEIMAGE\_PARTITION\_SIZE  := 0x0A000000
25. BOARD\_TVCONFIGIMAGE\_PARTITION\_SIZE   := 0x00A00000
26. BOARD\_TVDATABASEIMAGE\_PARTITION\_SIZE := 0x00800000
27. BOARD\_TVCUSTOMERIMAGE\_PARTITION\_SIZE := 0x01000000
29. BOARD\_FLASH\_BLOCK\_SIZE := 512

      （4）启动脚本加载，init.monet.rc脚本中on fs 模块中启动

**[plain]** [view plain](http://blog.csdn.net/zxh821112/article/details/50427759) [copy](http://blog.csdn.net/zxh821112/article/details/50427759)

1. on fs
2. mkdir /tvservice
3. mkdir /tvconfig
4. mkdir /tvdatabase
5. mkdir /tvcustomer
6. mount\_all /fstab.monet

      （5）init进程解析fstab.monet文件

         keywords.h文件中注册mount\_all操作方法

**[plain]** [view plain](http://blog.csdn.net/zxh821112/article/details/50427759) [copy](http://blog.csdn.net/zxh821112/article/details/50427759)

1. enum {
2. K\_UNKNOWN,
3. #endif
4. KEYWORD(capability,  OPTION,  0, 0)
5. KEYWORD(chdir,       COMMAND, 1, do\_chdir)
6. KEYWORD(chroot,      COMMAND, 1, do\_chroot)
7. KEYWORD(class,       OPTION,  0, 0)
8. KEYWORD(class\_start, COMMAND, 1, do\_class\_start)
9. KEYWORD(class\_stop,  COMMAND, 1, do\_class\_stop)
10. KEYWORD(class\_reset, COMMAND, 1, do\_class\_reset)
11. KEYWORD(console,     OPTION,  0, 0)
12. KEYWORD(critical,    OPTION,  0, 0)
13. KEYWORD(disabled,    OPTION,  0, 0)
14. KEYWORD(domainname,  COMMAND, 1, do\_domainname)
15. KEYWORD(enable,      COMMAND, 1, do\_enable)
16. KEYWORD(exec,        COMMAND, 1, do\_exec)
17. KEYWORD(export,      COMMAND, 2, do\_export)
18. KEYWORD(group,       OPTION,  0, 0)
19. KEYWORD(hostname,    COMMAND, 1, do\_hostname)
20. KEYWORD(ifup,        COMMAND, 1, do\_ifup)
21. KEYWORD(insmod,      COMMAND, 1, do\_insmod)
22. KEYWORD(import,      SECTION, 1, 0)
23. KEYWORD(keycodes,    OPTION,  0, 0)
24. KEYWORD(mkdir,       COMMAND, 1, do\_mkdir)
25. KEYWORD(mount\_all,   COMMAND, 1, do\_mount\_all)
26. KEYWORD(mount,       COMMAND, 3, do\_mount)
27. KEYWORD(on,          SECTION, 0, 0)
28. KEYWORD(oneshot,     OPTION,  0, 0)
29. KEYWORD(onrestart,   OPTION,  0, 0)
30. KEYWORD(powerctl,    COMMAND, 1, do\_powerctl)
31. KEYWORD(restart,     COMMAND, 1, do\_restart)
32. KEYWORD(restorecon,  COMMAND, 1, do\_restorecon)
33. KEYWORD(restorecon\_recursive,  COMMAND, 1, do\_restorecon\_recursive)
34. KEYWORD(rm,          COMMAND, 1, do\_rm)
35. KEYWORD(rmdir,       COMMAND, 1, do\_rmdir)
36. KEYWORD(seclabel,    OPTION,  0, 0)
37. KEYWORD(service,     SECTION, 0, 0)
38. KEYWORD(setcon,      COMMAND, 1, do\_setcon)
39. KEYWORD(setenforce,  COMMAND, 1, do\_setenforce)
40. KEYWORD(setenv,      OPTION,  2, 0)
41. KEYWORD(setkey,      COMMAND, 0, do\_setkey)
42. KEYWORD(setprop,     COMMAND, 2, do\_setprop)
43. KEYWORD(setrlimit,   COMMAND, 3, do\_setrlimit)
44. KEYWORD(setsebool,   COMMAND, 2, do\_setsebool)
45. KEYWORD(socket,      OPTION,  0, 0)
46. KEYWORD(start,       COMMAND, 1, do\_start)
47. KEYWORD(stop,        COMMAND, 1, do\_stop)
48. KEYWORD(swapon\_all,  COMMAND, 1, do\_swapon\_all)
49. KEYWORD(trigger,     COMMAND, 1, do\_trigger)
50. KEYWORD(symlink,     COMMAND, 1, do\_symlink)
51. KEYWORD(sysclktz,    COMMAND, 1, do\_sysclktz)
52. KEYWORD(user,        OPTION,  0, 0)
53. KEYWORD(wait,        COMMAND, 1, do\_wait)
54. KEYWORD(write,       COMMAND, 2, do\_write)
55. KEYWORD(copy,        COMMAND, 2, do\_copy)
56. KEYWORD(chown,       COMMAND, 2, do\_chown)
57. KEYWORD(chmod,       COMMAND, 2, do\_chmod)
58. KEYWORD(loglevel,    COMMAND, 1, do\_loglevel)
59. KEYWORD(load\_persist\_props,    COMMAND, 0, do\_load\_persist\_props)
60. KEYWORD(load\_all\_props,        COMMAND, 0, do\_load\_all\_props)
61. KEYWORD(ioprio,      OPTION,  0, 0)
62. #ifdef \_\_MAKE\_KEYWORD\_ENUM\_\_
63. KEYWORD\_COUNT,
64. }

         builtins.c文件中do\_mount\_all，启动一个进程调用fs\_mgr\_read\_fstab，fs\_mgr\_mount\_all启动解析，其中system\core\fs\_mgr有被vold，adb等模块用来解析

         fstab.xxx等分区文件。

       （6）根据第三步中指定的分区大小，进行打包对应分区的img文件

          releaseimage.sh, mkimage.sh, build\_image.sh, make\_usb\_upgrade.sh等脚本中，会根据第三步设置的大小来制作img文件

          如mkimage.sh

**[plain]** [view plain](http://blog.csdn.net/zxh821112/article/details/50427759) [copy](http://blog.csdn.net/zxh821112/article/details/50427759)

1. function make-images()
2. {
3. mkdir -p $RELEASE\_OUT
4. mkdir -p $RELEASE\_OUT/scripts
6. gernerate-script-start
8. # Raw partition must be first.
9. make-raw-partition
10. make-split-fs-partition system $BOARD\_SYSTEMIMAGE\_PARTITION\_SIZE true
11. make-fs-partition userdata $BOARD\_USERDATAIMAGE\_PARTITION\_SIZE true
12. make-fs-partition cache $BOARD\_CACHEIMAGE\_PARTITION\_SIZE true
13. make-fs-partition tvservice $BOARD\_TVSERVICEIMAGE\_PARTITION\_SIZE false
14. make-fs-partition tvconfig $BOARD\_TVCONFIGIMAGE\_PARTITION\_SIZE false
15. make-fs-partition tvdatabase $BOARD\_TVDATABASEIMAGE\_PARTITION\_SIZE false
16. make-fs-partition tvcustomer $BOARD\_TVCUSTOMERIMAGE\_PARTITION\_SIZE false
17. ......
18. ......
20. }

           buiild\_image.sh文件

**[plain]** [view plain](http://blog.csdn.net/zxh821112/article/details/50427759) [copy](http://blog.csdn.net/zxh821112/article/details/50427759)

1. #make-fs-partition $PARTITION\_NAME $PARTITION\_SIZE $PARTITION\_LZO
2. function make-fs-partition()
3. {
4. echo -e "\033[31mMake fs $1 partition...\033[0m"
5. local PARTITION\_NAME=$1
6. local PARTITION\_SIZE=$2
7. local PARTITION\_LZO=$3
9. # Generate image
10. gernerate-fs-image $PARTITION\_NAME $PARTITION\_SIZE $PARTITION\_LZO
12. release-fs-partition $PARTITION\_NAME $PARTITION\_SIZE $PARTITION\_LZO
13. }

**[plain]** [view plain](http://blog.csdn.net/zxh821112/article/details/50427759) [copy](http://blog.csdn.net/zxh821112/article/details/50427759)

1. function gernerate-fs-image()
2. {
3. local PARTITION\_NAME=$1
4. local PARTITION\_SIZE=$2
5. local PARTITION\_LZO=$3
7. if [ "$PARTITION\_NAME" == "userdata" ]; then
8. local MOUNT\_POINT=data
9. else
10. local MOUNT\_POINT=$PARTITION\_NAME
11. fi
13. local TARGET\_OUT=$PRODUCT\_OUT/$MOUNT\_POINT
14. if [ ! -d "$TARGET\_OUT" ]; then
15. echo "No $TARGET\_OUT directory to gernerate image."
16. return
17. fi
19. if [ "$PARTITION\_NAME" == "system" -o "$PARTITION\_NAME" == "userdata" -o "$PARTITION\_NAME" == "cache" -o "$PARTITION\_NAME" == "systembackup" ]; then
20. if [  $PARTITION\_NAME == "system" ]; then
21. local IMAGE\_INFO=$PRODUCT\_OUT/obj/PACKAGING/systemimage\_intermediates/system\_image\_info.txt
22. else
23. local IMAGE\_INFO=$PRODUCT\_OUT/obj/PACKAGING/"$PARTITION\_NAME"\_intermediates/"$PARTITION\_NAME"\_image\_info.txt
24. fi
26. generate-userimage-prop-dictionary $IMAGE\_INFO
27. ./build/tools/releasetools/build\_image.py $TARGET\_OUT $IMAGE\_INFO $PRODUCT\_OUT/$PARTITION\_NAME.img
28. else
29. <span style="background-color: rgb(0, 102, 0);">make\_ext4fs -S $PRODUCT\_OUT/root/file\_contexts -l $PARTITION\_SIZE -a $MOUNT\_POINT $PRODUCT\_OUT/$PARTITION\_NAME.img $TARGET\_OUT</span>
30. fi
31. }

          （7）最后make\_use\_upgrade.sh打包，发布整体软件。