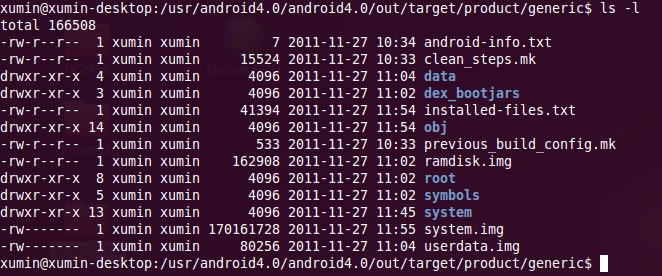
**1、Android文件系统的结构**

官方android源码编译后得到的文件如下：

其中有system.img,ramdisk.img,userdata.img三个映像文件。其中，ramdisk.img是文件系统，system.img包括了主要的包、库等文件，userdata.img包括了一些用户数据，当emulator加载这3个映像文件后，会把system.img和userdata.img分别加载到ramdisk文件系统中的system和userdata目录下。  
  
**2、分离组建Android文件系统**  
system.img,ramdisk.img,userdata.img映像文件是采用cpio打包、gzip压缩的，可以通过file命令验证：  
filer amdisk.img，输出：

**[plain]** [view plain](http://blog.csdn.net/conowen/article/details/7251057) [copy](http://blog.csdn.net/conowen/article/details/7251057)

1. ramdisk.img:gzip compressed data, from Unix, last modified:

其中ramdisk.img是一个分区映像文件，即是一个 文件系统（root file system），，kernel启动时，ramdisk负责init，system.img包括了主要系统，系统app等，挂载于文件系统下的 system文件夹, userdata.img包括了一些用户数据，被挂载到文件系统下的data文件夹下。

另外Android源码编译后还有root、system、data三个文件夹，其实就是上述三个img解压之后的内容而已，root文件夹里面的内容就是ramdisk.img解压后的内容，即是一个文件系统，包括system、data、bin、dev等目录。而system与data则对应着system.img与userdata.img的解压后的内容。

验证一下就行了，将ramdisk.img复制一份到任何其他目录下，将其名称改为ramdisk.img.gz，并使用命令

**[java]** [view plain](http://blog.csdn.net/conowen/article/details/7251057) [copy](http://blog.csdn.net/conowen/article/details/7251057)

1. gunzip ramdisk.img.gz

然后新建一个名为ramdisk的文件夹，转到ramdisk目录，输入命令

**[html]** [view plain](http://blog.csdn.net/conowen/article/details/7251057) [copy](http://blog.csdn.net/conowen/article/details/7251057)

1. <pre name="code" class="java">cpio-i -F ../ramdisk.img

这下，就能看见ramdisk里面的内容与root文件夹一样的了。然后把Android源码编译后生成的system和userdata里的文件复制到ramdisk/system和ramdisk/userdata下。这样就得到一个完整的文件系统了。

=====================================================================================================

**附：Android文件系统(rootfs)**（以下内容为转载）

**Android的文件系统根目录的结构**

Android的文件系统根目录的结构：

**[java]** [view plain](http://blog.csdn.net/conowen/article/details/7251057) [copy](http://blog.csdn.net/conowen/article/details/7251057)

1. drwxrwxrwt  root     root              2009-06-10 09:53 sqlite\_stmt\_journals
2. drwxrwx--- system   cache             2008-09-06 22:51 cache
3. d---rwxrwx system   system            1970-01-01 08:00 sdcard
4. lrwxrwxrwx root     root              2009-06-09 22:11 etc -> /system/etc
5. drwxr-xr-x root     root              2008-09-06 22:45 system
6. drwxr-xr-x root     root              1970-01-01 08:00 sys
7. drwxr-x--- root     root              1970-01-01 08:00 sbin
8. -rw-r--r-- root     root          117 1970-01-01 08:00 runme.sh
9. dr-xr-xr-x root     root              1970-01-01 08:00 proc
10. -rwxr-x--- root     root         1704 1970-01-01 08:00 init.trout.rc
11. -rwxr-x--- root     root         9329 1970-01-01 08:00 init.rc
12. -rwxr-x--- root     root         1677 1970-01-01 08:00 init.goldfish.rc
13. -rwxr-x--- root     root       106636 1970-01-01 08:00 init
14. -rw-r--r-- root     root          118 1970-01-01 08:00 default.prop
15. drwxrwx--x system   system            2008-09-06 22:51 data
16. drwx------ root     root              2009-06-07 16:29 root
17. drwxr-xr-x root     root              2009-06-09 22:11 dev

**sqlite\_stmt\_journals**:一个根目录下的tmpfs文件系统，用于存放临时文件数据。

**cache** ： 是缓存临时文件夹，据说是除了T-mobile的OTA更新外，别无用处。  
**sdcard**：是SD卡中的FAT32文件系统挂载的目录  
**etc**  ：指向 /system/etc ，众所周知的配置文件存放目录

**system** ：是一个很重要的目录，系统中的大部分东西都在这里了。

**sys** ：用于挂载 sysfs文件系统。 在设备模型中,sysfs文件系统用来表示设备的结构.将设备的层次结构形象的反应到用户空间中.用户空间可以修改sysfs中的文件属性来修改设备的属性值**sbin**： 只放了一个用于调试的adbd程序。

**proc**：**/proc** 文件系统下的多种文件提供的系统信息不是针对某个特定进程的，而是能够在整个系统范围的上下文中使用。  
**data**：存放用户安装的软件以及各种数据。  
**root** ：什么都没有。

**dev**：不用多说了，设备节点文件的存放地。

**下面介绍非目录的文件：**  
  
runme.sh用于 SD 卡中 EXT2文件系统的自动挂载动作的脚本。  
init.trout.rc，init.rc，init.goldfish.rc 是初始化文件.  
init是系统启动到文件系统的时候第一个运行的程序。  
  
  
从以上的根目录分析来看，Android的根文件系统并非标准的Linux文件系统，所以以后还得仔细分析一下启动过程，才能认识Android系统。  
今天要来分析一下Android文件系统的/system目录的结构。

着重介绍一下/system目录

/system目录是在Android文件系统占有及其重要的位置，基本上所有的工具和应用程序都在这个目录下，我看来是一个真正rootfs。他在Android手机中存放在nandflash的mtd3中，是一个yaffs2文件系统，在启动时被挂载在root的/system目录下，其中包含有：

 下面逐个分析其中的目录：

**[java]** [view plain](http://blog.csdn.net/conowen/article/details/7251057) [copy](http://blog.csdn.net/conowen/article/details/7251057)

1. # ls -a -l /system
2. drwxr-xr-x root     208               1970-01-01 08:00 xbin
3. drwxr-xr-x root     root              1970-01-01 08:00 modules
4. drwxr-xr-x root     root              2008-08-01 20:00 framework
5. drwxr-xr-x root     root              2008-08-01 20:00 fonts
6. drwxr-xr-x root     root              2008-08-01 20:00 etc
7. -rw-r--r-- root     root         2197 2008-08-01 20:00 build.prop
8. drwxr-xr-x root     root              2008-08-01 20:00 media
9. drwxr-xr-x root     shell             2008-08-01 20:00 bin
10. drwxr-xr-x root     root              2008-08-01 20:00 usr
11. drwxr-xr-x root     root              2008-08-01 20:00  app
12. drwxr-xr-x root     root              2008-09-06 22:45 lost+found
13. drwxr-xr-x root     root              2008-08-01 20:00 lib
14. drwxr-xr-x root     root              2008-08-01 20:00 sd
15. -rw-r--r-- root     root          145 2008-08-01 20:00 init.rc

**xbin**  ：下放了很多系统管理工具，这些工具不是到toolbox的链接，每个都是可执行程序。如果你看 到这些命令你会发现他们根本   不常用，他们都是为系统管理员准备的，是一些系统管理和配置工具。这个文件夹的作用相当于标准Linux文件系统中的 /sbin。我的手机此目录下有busybox，肯定是改过的，应该是破解者加上的。  
**modules**：使用来存放内核模块（主要是fs和net）和模块配置文件的地方。  
**framework**： 是JAVA平台的一些核心文件，属于JAVA平台系统框架文件。里面的文件都是.jar和.odex文件。  
                   备注：什么是odex文件？ odex是被优化过的JAVA程序文件，体积通常是.jar的4倍左右。执行效率比.jar高。  
**fonts**      ：很显然，这是字体库文件的存放目录。  
**etc**       ：这里存放了系统中几乎所有的配置文件，根目录下的/etc就链结于此。  
**build.prop**  ：是一个属性文件，在Android系统中.prop文件很重要，记录了系统的设置和改变，类似于/etc中的文件。  
**media**      ：里面主要是存放了系统的铃声的，分为 notifications（通知）、ui（界面）、alarms（警告）和ringtones（铃声），里面都是.ogg音频文件。  
**bin**：众所周知，是存放用户常用的工具程序的，其中大部分是到toolbox的链接（类似嵌入式Linux中的busybox）。toolbox应该是google简化版的busybox，我还没深入研究过。  
**usr**        ：用户的配置文件，如键盘布局、共享、时区文件等等。您可以cat 来看看。  
**app**          ：顾名思义，存放的是Android系统自带的JAVA应用程序。  
**lost+found** ：yaffs文件系统固有的，类似回收站的文件夹，只有是yaffs文件系统都会有。  
**lib**           ：存放几乎所有的共享库（.so）文件。  
**sd**          ：SD卡中的EXT2分区的挂载目录  
**init.rc**       :一个初始化脚本，用于将/system/modules和/system/xbin挂载为cramfs，避免系统被无意破坏。