\system\app   
这个里面主要存放的是常规下载的应用程序，可以看到都是以APK格式结尾的文件。在这个文件夹下的程序为系统默认的组件，自己安装的软件将不会出现在这里，而是\data\文件夹中。下面是详细的介绍:  
\system\app\AlarmClock.apk 闹钟  
\system\app\AlarmClock.odex  
\system\app\Browser.apk 浏览器  
\system\app\Browser.odex  
\system\app\Bugreport.apk Bug报告  
\system\app\Bugreport.odex  
\system\app\Calculator.apk 计算器  
\system\app\Calculator.odex  
\system\app\Calendar.apk 日历  
\system\app\Calendar.odex  
\system\app\CalendarProvider.apk 日历提供  
\system\app\CalendarProvider.odex  
\system\app\Camera.apk 照相机  
\system\app\Camera.odex  
\system\app\com.amazon.mp3.apk 亚马逊音乐  
\system\app\Contacts.apk 联系人  
\system\app\Contacts.odex  
\system\app\DownloadProvider.apk 下载提供  
\system\app\DownloadProvider.odex  
\system\app\DrmProvider.apk DRM数字版权提供  
\system\app\DrmProvider.odex  
\system\app\Email.apk 电子邮件客户端  
\system\app\Email.odex  
\system\app\FieldTest.apk 测试程序  
\system\app\FieldTest.odex  
\system\app\GDataFeedsProvider.apk GoogleData提供  
\system\app\GDataFeedsProvider.odex  
\system\app\Gmail.apk Gmail电子邮件  
\system\app\Gmail.odex  
\system\app\GmailProvider.apk Gmail提供  
\system\app\GmailProvider.odex  
\system\app\GoogleApps.apk 谷歌程序包  
\system\app\GoogleApps.odex  
\system\app\GoogleSearch.apk 搜索工具  
\system\app\GoogleSearch.odex  
\system\app\gtalkservice.apk GTalk服务  
\system\app\gtalkservice.odex  
\system\app\HTMLViewer.apk HTML查看器  
\system\app\HTMLViewer.odex  
\system\app\IM.apk 即使通讯组件包含MSN、yahoo通  
\system\app\ImCredentialProvider.apk  
\system\app\ImProvider.apk  
\system\app\ImProvider.odex  
\system\app\Launcher.apk 启动加载器  
\system\app\Launcher.odex  
\system\app\Maps.apk 电子地图  
\system\app\Maps.odex  
\system\app\MediaProvider.apk 多媒体播放提供  
\system\app\MediaProvider.odex  
\system\app\Mms.apk 短信、彩信  
\system\app\Mms.odex  
\system\app\Music.apk 音乐播放器  
\system\app\Music.odex  
\system\app\MyFaves.apk T-Mobile MyFaves程序  
\system\app\MyFaves.odex  
\system\app\PackageInstaller.apk apk安装程序  
\system\app\PackageInstaller.odex  
\system\app\Phone.apk 电话拨号器  
\system\app\Phone.odex  
\system\app\Settings.apk 系统设置  
\system\app\Settings.odex  
\system\app\SettingsProvider.apk 设置提供  
\system\app\SettingsProvider.odex  
\system\app\SetupWizard.apk 设置向导  
\system\app\SetupWizard.odex  
\system\app\SoundRecorder.apk 录音工具  
\system\app\SoundRecorder.odex  
\system\app\Street.apk 街景地图  
\system\app\Street.odex  
\system\app\Sync.apk 同步程序  
\system\app\Sync.odex  
\system\app\Talk.apk 语音程序  
\system\app\Talk.odex  
\system\app\TelephonyProvider.apk 电话提供  
\system\app\TelephonyProvider.odex  
\system\app\Updater.apk 更新程序  
\system\app\Updater.odex  
\system\app\Vending.apk 制造商信息  
\system\app\Vending.odex  
\system\app\VoiceDialer.apk 语音拨号器  
\system\app\VoiceDialer.odex  
\system\app\YouTube.apk Youtube视频  
\system\app\YouTube.odex

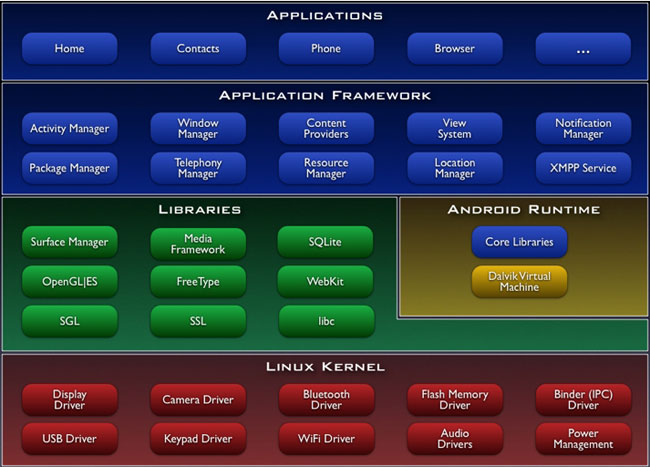
\system\bin  
这个目录下的文件都是系统的本地程序，从bin文件夹名称可以看出是binary二进制的程序，里面主要是Linux系统自带的组件，Android手机网就主要文件做下简单的分析介绍：  
\system\bin\akmd  
\system\bin\am  
\system\bin\app\_process 系统进程  
\system\bin\dalvikvm Dalvik虚拟机宿主  
\system\bin\dbus-daemon 系统BUS总线监控  
\system\bin\debuggerd 调试器  
\system\bin\debug\_tool 调试工具  
\system\bin\dexopt DEX选项  
\system\bin\dhcpcd DHCP服务器  
\system\bin\dumpstate 状态抓取器  
\system\bin\dumpsys 系统抓取器  
\system\bin\dvz  
\system\bin\fillup  
\system\bin\flash\_image 闪存映像  
\system\bin\hciattach  
\system\bin\hcid HCID内核  
\system\bin\hostapd  
\system\bin\hostapd\_cli  
\system\bin\htclogkernel  
\system\bin\input  
\system\bin\installd  
\system\bin\itr  
\system\bin\linker  
\system\bin\logcat Logcat日志打印  
\system\bin\logwrapper  
\system\bin\mediaserver  
\system\bin\monkey  
\system\bin\mountd 存储挂载器  
\system\bin\netcfg 网络设置  
\system\bin\ping Ping程序  
\system\bin\playmp3 MP3播放器  
\system\bin\pm 包管理器  
\system\bin\qemud QEMU虚拟机  
\system\bin\radiooptions 无线选项  
\system\bin\rild RIL组件  
\system\bin\sdptool  
\system\bin\sdutil  
\system\bin\service   
\system\bin\servicemanager 服务管理器  
\system\bin\sh  
\system\bin\ssltest SSL测试  
\system\bin\surfaceflinger 触摸感应驱动  
\system\bin\svc 服务  
\system\bin\system\_server  
\system\bin\telnetd Telnet组件  
\system\bin\toolbox  
\system\bin\wlan\_loader  
\system\bin\wpa\_cli  
\system\bin\wpa\_supplicant  
\system\etc  
从文件夹名称来看保存的都是系统的配置文件，比如APN接入点设置等核心配置。  
\system\etc\apns-conf.xml APN接入点配置文件  
\system\etc\AudioFilter.csv 音频过滤器配置文件  
\system\etc\AudioPara4.csv  
\system\etc\bookmarks.xml 书签数据库  
\system\etc\dbus.conf 总线监视配置文件  
\system\etc\dhcpcd  
\system\etc\event-log-tags  
\system\etc\favorites.xml 收藏夹  
\system\etc\firmware 固件信息  
\system\etc\gps.conf GPS设置文件  
\system\etc\hcid.conf  内核HCID配置文件  
\system\etc\hosts 网络DNS缓存  
\system\etc\init.goldfish.sh  
\system\etc\location 定位相关  
\system\etc\mountd.conf 存储挂载配置文件  
\system\etc\NOTICE.html 提示网页  
\system\etc\permissions.xml 权限许可  
\system\etc\pvplayer.conf   
\system\etc\security   
\system\etc\wifi WLAN相关组件  
\system\etc\dhcpcd\dhcpcd-hooks  
\system\etc\dhcpcd\dhcpcd-run-hooks  
\system\etc\dhcpcd\dhcpcd.conf  
\system\etc\dhcpcd\dhcpcd-hooks\01-test  
\system\etc\dhcpcd\dhcpcd-hooks\20-dns.conf  
\system\etc\dhcpcd\dhcpcd-hooks\95-configured  
\system\etc\firmware\brf6300.bin  
\system\etc\location\gps  
[page\_break]   
\system\etc\location\gps\location 定位相关  
\system\etc\location\gps\nmea GPS数据解析  
\system\etc\location\gps\properties  
\system\etc\security\cacerts.bks  
\system\etc\security\otacerts.zip OTA下载验证  
\system\etc\wifi\Fw1251r1c.bin  
\system\etc\wifi\tiwlan.ini  
\system\etc\wifi\wpa\_supplicant.conf WPA验证组件  
\system\fonts  
字体文件夹，除了标准字体和粗体、斜体外可以看到文件体积最大的可能是中文字库，或一些unicode字库，从T-Mobile G1上可以清楚的看到显示简体中文正常，其中DroidSansFallback.ttf文件大小  
\system\fonts\DroidSans-Bold.ttf  
\system\fonts\DroidSans.ttf  
\system\fonts\DroidSansFallback.ttf  
\system\fonts\DroidSansMono.ttf  
\system\fonts\DroidSerif-Bold.ttf  
\system\fonts\DroidSerif-BoldItalic.ttf  
\system\fonts\DroidSerif-Italic.ttf  
\system\fonts\DroidSerif-Regular.ttf  
\system\framework  
framework主要是一些核心的文件，从后缀名为jar可以看出是是系统平台框架。  
\system\framework\am.jar  
\system\framework\am.odex  
\system\framework\android.awt.jar AWT库  
\system\framework\android.awt.odex  
\system\framework\android.policy.jar   
\system\framework\android.policy.odex  
\system\framework\android.test.runner.jar  
\system\framework\android.test.runner.odex  
\system\framework\com.google.android.gtalkservice.jar GTalk服务  
\system\framework\com.google.android.gtalkservice.odex  
\system\framework\com.google.android.maps.jar 电子地图库  
\system\framework\com.google.android.maps.odex  
\system\framework\core.jar 核心库，启动桌面时首先加载这个  
\system\framework\core.odex  
\system\framework\ext.jar  
\system\framework\ext.odex  
\system\framework\framework-res.apk  
\system\framework\framework-tests.jar  
\system\framework\framework-tests.odex  
\system\framework\framework.jar  
\system\framework\framework.odex  
\system\framework\input.jar 输入库  
\system\framework\input.odex  
\system\framework\itr.jar  
\system\framework\itr.odex  
\system\framework\monkey.jar  
\system\framework\monkey.odex  
\system\framework\pm.jar 包管理库  
\system\framework\pm.odex  
\system\framework\services.jar  
\system\framework\services.odex  
\system\framework\ssltest.jar  
\system\framework\ssltest.odex  
\system\framework\svc.jar 系统服务  
\system\framework\svc.odex

\system\lib  
  
lib目录中存放的主要是系统底层库，如平台运行时库。  
\system\lib\libaes.so   
\system\lib\libagl.so  
\system\lib\libandroid\_runtime.so Android运行时库  
\system\lib\libandroid\_servers.so 系统服务组件  
\system\lib\libaudio.so 音频处理  
\system\lib\libaudioeq.so EQ均衡器  
\system\lib\libaudioflinger.so 音频过滤器  
\system\lib\libbluetooth.so 蓝牙组件  
\system\lib\libc.so  
\system\lib\libcamera.so 超相机组件  
\system\lib\libcameraservice.so  
\system\lib\libcorecg.so  
\system\lib\libcrypto.so 加密组件  
\system\lib\libctest.so  
\system\lib\libcutils.so  
\system\lib\libdbus.so  
\system\lib\libdl.so  
\system\lib\libdrm1.so DRM解析库  
\system\lib\libdrm1\_jni.so  
\system\lib\libdvm.so  
\system\lib\libexif.so  
\system\lib\libexpat.so  
\system\lib\libFFTEm.so  
\system\lib\libGLES\_CM.so  
\system\lib\libgps.so  
\system\lib\libhardware.so  
\system\lib\libhgl.so  
\system\lib\libhtc\_ril.so  
\system\lib\libicudata.so  
\system\lib\libicui18n.so  
\system\lib\libicuuc.so  
\system\lib\liblog.so  
\system\lib\libm.so  
\system\lib\libmedia.so  
\system\lib\libmediaplayerservice.so  
\system\lib\libmedia\_jni.so  
\system\lib\libnativehelper.so  
\system\lib\libnetutils.so  
\system\lib\libOmxCore.so  
\system\lib\libOmxH264Dec.so  
\system\lib\libpixelflinger.so  
\system\lib\libpvasf.so  
\system\lib\libpvasfreg.so  
\system\lib\libpvauthor.so  
\system\lib\libpvcommon.so  
\system\lib\libpvdownload.so  
\system\lib\libpvdownloadreg.so  
\system\lib\libpvmp4.so  
\system\lib\libpvmp4reg.so  
\system\lib\libpvnet\_support.so  
\system\lib\libpvplayer.so  
\system\lib\libpvrtsp.so  
\system\lib\libpvrtspreg.so  
\system\lib\libqcamera.so  
\system\lib\libreference-ril.so  
\system\lib\libril.so  
\system\lib\librpc.so  
\system\lib\libsgl.so  
\system\lib\libsonivox.so  
\system\lib\libsoundpool.so  
\system\lib\libsqlite.so  
\system\lib\libssl.so  
\system\lib\libstdc++.so  
\system\lib\libsurfaceflinger.so  
\system\lib\libsystem\_server.so  
\system\lib\libthread\_db.so  
\system\lib\libUAPI\_jni.so  
\system\lib\libui.so  
\system\lib\libutils.so  
\system\lib\libvorbisidec.so  
\system\lib\libwbxml.so  
\system\lib\libwbxml\_jni.so  
\system\lib\libwebcore.so  
\system\lib\libwpa\_client.so  
\system\lib\libxml2wbxml.so  
\system\lib\libz.so  
\system\lib\modules  
\system\lib\modules\wlan.ko  
\system\media  
[page\_break]   
铃声音乐文件夹，除了常规的铃声外还有一些系统提示事件音  
\system\media\audio  
\system\media\audio\alarms 闹铃音  
\system\media\audio\notifications 提示音  
\system\media\audio\ringtones 铃声  
\system\media\audio\ui 界面操作事件音  
\system\media\audio\alarms\Alarm\_Beep\_01.ogg  
\system\media\audio\alarms\Alarm\_Beep\_02.ogg  
\system\media\audio\alarms\Alarm\_Beep\_03.ogg  
\system\media\audio\alarms\Alarm\_Buzzer.ogg  
\system\media\audio\alarms\Alarm\_Classic.ogg  
\system\media\audio\alarms\Alarm\_Rooster\_02.ogg  
\system\media\audio\notifications\Beat\_Box\_Android.ogg  
\system\media\audio\notifications\CaffeineSnake.ogg  
\system\media\audio\notifications\DearDeer.ogg  
\system\media\audio\notifications\DontPanic.ogg  
\system\media\audio\notifications\F1\_MissedCall.ogg  
\system\media\audio\notifications\F1\_New\_MMS.ogg  
\system\media\audio\notifications\F1\_New\_SMS.ogg  
\system\media\audio\notifications\Heaven.ogg  
\system\media\audio\notifications\Highwire.ogg  
\system\media\audio\notifications\KzurbSonar.ogg  
\system\media\audio\notifications\OnTheHunt.ogg  
\system\media\audio\notifications\TaDa.ogg  
\system\media\audio\notifications\Tinkerbell.ogg  
\system\media\audio\notifications\Voila.ogg  
\system\media\audio\ringtones\BeatPlucker.ogg  
\system\media\audio\ringtones\BentleyDubs.ogg  
\system\media\audio\ringtones\BirdLoop.ogg  
\system\media\audio\ringtones\CaribbeanIce.ogg  
\system\media\audio\ringtones\CrazyDream.ogg  
\system\media\audio\ringtones\CurveBall.ogg  
\system\media\audio\ringtones\DreamTheme.ogg  
\system\media\audio\ringtones\EtherShake.ogg  
\system\media\audio\ringtones\FriendlyGhost.ogg  
\system\media\audio\ringtones\GameOverGuitar.ogg  
\system\media\audio\ringtones\Growl.ogg  
\system\media\audio\ringtones\InsertCoin.ogg  
\system\media\audio\ringtones\LoopyLounge.ogg  
\system\media\audio\ringtones\LoveFlute.ogg  
\system\media\audio\ringtones\MidEvilJaunt.ogg  
\system\media\audio\ringtones\MildlyAlarming.ogg  
\system\media\audio\ringtones\NewPlayer.ogg  
\system\media\audio\ringtones\Noises1.ogg  
\system\media\audio\ringtones\Noises2.ogg  
\system\media\audio\ringtones\Noises3.ogg  
\system\media\audio\ringtones\OrganDub.ogg  
\system\media\audio\ringtones\Ring\_Classic\_02.ogg  
\system\media\audio\ringtones\Ring\_Digital\_02.ogg  
\system\media\audio\ringtones\Ring\_Synth\_02.ogg  
\system\media\audio\ringtones\Ring\_Synth\_04.ogg  
\system\media\audio\ringtones\RomancingTheTone.ogg  
\system\media\audio\ringtones\SitarVsSitar.ogg  
\system\media\audio\ringtones\SpringyJalopy.ogg  
\system\media\audio\ringtones\T-Jingle.ogg  
\system\media\audio\ringtones\Terminated.ogg  
\system\media\audio\ringtones\TwirlAway.ogg  
\system\media\audio\ringtones\VeryAlarmed.ogg  
\system\media\audio\ringtones\World.ogg  
\system\media\audio\ui\Effect\_Tick.ogg  
\system\sounds  
默认的音乐测试文件，仅有一个test.mid文件，用于播放测试的文件。  
\system\sounds\test.mid  
\system\usr  
用户文件夹，包含共享、键盘布局、时间区域文件等。  
\system\usr\keychars  
\system\usr\keylayout  
\system\usr\share  
\system\usr\srec  
\system\usr\keychars\qwerty.kcm.bin  
\system\usr\keychars\qwerty2.kcm.bin  
\system\usr\keychars\trout-keypad-qwertz.kcm.bin  
\system\usr\keychars\trout-keypad-v2.kcm.bin  
\system\usr\keychars\trout-keypad-v3.kcm.bin  
\system\usr\keychars\trout-keypad.kcm.bin  
\system\usr\keylayout\h2w\_headset.kl  
\system\usr\keylayout\qwerty.kl  
\system\usr\keylayout\trout-keypad-qwertz.kl  
\system\usr\keylayout\trout-keypad-v2.kl  
\system\usr\keylayout\trout-keypad-v3.kl  
\system\usr\keylayout\trout-keypad.kl  
\system\usr\share\bsk  
\system\usr\share\zoneinfo  
\system\usr\share\bsk\V\_FD\_speed\_101.bsk  
\system\usr\share\bsk\V\_FD\_std\_101.bsk  
\system\usr\share\zoneinfo\zoneinfo.dat  
\system\usr\share\zoneinfo\zoneinfo.idx  
\system\usr\srec\config  
\system\usr\srec\config\en.us  
\system\usr\srec\config\en.us\baseline.par  
\system\usr\srec\config\en.us\baseline11k.par  
\system\usr\srec\config\en.us\baseline8k.par  
\system\usr\srec\config\en.us\dictionary  
\system\usr\srec\config\en.us\g2p  
\system\usr\srec\config\en.us\grammars  
\system\usr\srec\config\en.us\models  
\system\usr\srec\config\en.us\dictionary\basic.ok  
\system\usr\srec\config\en.us\dictionary\cmu6plus.ok.zip  
\system\usr\srec\config\en.us\dictionary\enroll.ok  
\system\usr\srec\config\en.us\g2p\en-US-ttp.data  
\system\usr\srec\config\en.us\grammars\VoiceDialer.g2g  
\system\usr\srec\config\en.us\models\generic.swiarb  
\system\usr\srec\config\en.us\models\generic11.lda  
\system\usr\srec\config\en.us\models\generic11\_f.swimdl  
\system\usr\srec\config\en.us\models\generic11\_m.swimdl  
\system\usr\srec\config\en.us\models\generic8.lda  
\system\usr\srec\config\en.us\models\generic8\_f.swimdl  
\system\usr\srec\config\en.us\models\generic8\_m.swimdl  
整个Android平台的文件不止是这么多，部分文件在/data文件夹中都是用户文件夹，这里就不介绍了。

Android Internal - system directories  
您應該已經知道，Android 的核心作業系統是 Linux (現在用的版本是 2.6.25)。因此內部的檔案系統，與系統目錄等，也和 Linux 脫不了關係。要如何觀察這些系統目錄，到底藏了什麼秘密呢？  
在 Eclipse 環境中，你可以用 DDMS 來觀察。不過這個 DDMS 在我的電腦上的執行速度，實在太慢了。我個人的偏好是，直接用 adb shell 來觀察。啟動 adb shell 的用法如下：1. 在 Windows 中，開啟一個命令列視窗 （或執行 cmd.exe 這個程式）  
2. cd <android-sdk-install\_path>/tools  
3. adb shell  
  
當你看到這個 # 提示字元時，就表示你已經進入模擬器的系統。接下來，你就可以用 Linux 上的指令來瀏覽這些目錄。像是 cd, ls, pwd, cat, rm 等等。  
這裡面，有幾個目錄是和 Android 相關的，我特地將他整理下來，供你參考。**/system/app**   
預載入應用程式執行檔 (\*.apk)，都是放在這。像是 Alarm Clock, Browser, Contacts, Maps,... 等等。  
用 DDMS 觀察的話，目錄內的內容，就像下面這樣：  
**/system/framework**   
這會放 Android 系統的核心程式庫。像是 core.jar, framework-res.apk, com.google.android.gtalkservice.jar,...等等。疑，1.0r1 不是已經將 gtalk 等相關 APIs 移除了嗎？怎麼他的程式庫還在？雖然許多程式庫都是以 jar 結尾的，不過裡面 Java classes 還是以 dex 格式存在著。**/system/media/audio/(notification, alarms, ringtones, ui)**   
這裡放系統的聲音檔，像是鬧鈴聲，來電鈴聲等等。這些聲音檔，多是 ogg 格式。**/data/anr/traces.txt**   
當你的應用程式發生 ANR (Application is Not Responding) 錯誤時，Android 會自動將問題點的 code stack list 寫在這個檔案內，你直接用 cat 命令就可以看他的內容。**/data/app**   
/system/app 放的是系統預載的應用程式執行檔。而這裡放的是使用者自己安裝的應用程式執行檔 (\*.apk)。**/data/data/<app-package-name>**   
當你在程式中用 Context.openFileOutput() 所建立的檔案，都放在這個目錄下的 files 子目錄內。而用 Context.getSharedPreferences() 所建立的 preferences 檔 (\*.xml) ，則是放在 shared\_pref 這個子目錄中。**/data/location/gps**   
這裡看起來，應該是給 GPS location provider 用的。其中的 properties 檔案的內容如下：  
requiresNetwork false  
requiresSatellite false  
requiresCell false  
hasMonetaryCost false  
supportsAltitude true  
supportsBearing true  
supportsSpeed true  
repeat true  
accuracy 100  
powerRequirement 1  
  
沒猜錯的話，這應該是用來描述 GPS location provider 有那些功能的。**/data/system/location/location.gps**   
這個檔案也是個一般文字檔。主要是記錄最後的經緯度座標。LocationManager.getLastKnownLocation() 就在來這抓值的。**/data/property/persist.sys.timezone**   
這個檔案也是個一般文字檔。主要是記錄目前系統所使用的時區。在我的模擬器上，他記錄著 Asia/Taipei 這個字串。  
最後，下面這兩個指令，也很好用。他可以幫你將外部檔案，複製到模擬器的系統檔案目錄內，也可以將模擬器內的檔案，複製到外部系統上。adb push <local> <remote> - copy file/dir to device  
adb pull <remote> <local> - copy file/dir from device

Android是google公司针对手机开发的一个平台，并公布了其中大部分代码，其大部分应用程序都是用JAVA开发的，毕 竟它是商业性的产品嘛，有所保留也是理所当然的。对于搞嵌入式linux开发的人来说我们可以从中学习其长处，也算得上是未来的一个发展方向和趋势吧。

我们先来看看Android它的体系结构吧。下面是一张公开的Android体系结构图。



android的系统架构和其操作系统一样，采用了分层的架构。从架构图看，android分为四个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和linux核心层。蓝色的代表java程序，黄色的代码为运行JAVA程序而实现的虚拟机，绿色部分为C/C++语言编写的程序库，红色的代码内核(linux内核+drvier)

1.应用程序

Android会同一系列核心应用程序包一起发布，该应用程序包包括email客户端，SMS短消息程序，日历，地图，浏览器，联系人管理程序等。所有的应用程序都是使用JAVA语言编写的。

2.应用程序框架

开 发人员也可以完全访问核心应用程序所使用的API框架。该应用程序的架构设计简化了组件的重用；任何一个应用程序都可以发布它的功能块并且任何其 它的应用程序都可以使用其所发布的功能块（不过得遵循框架的安全性限制）。同样，该应用程序重用机制也使用户可以方便的替换程序组件。

隐藏在每个应用后面的是一系列的服务和系统, 其中包括；

    \* 丰富而又可扩展的视图（Views），可以用来构建应用程序， 它包括列表（lists），网格（grids），文本框（text boxes），按钮（buttons）， 甚至可嵌入的web浏览器。  
    \* 内容提供器（Content Providers）使得应用程序可以访问另一个应用程序的数据（如联系人数据库）， 或者共享它们自己的数据  
    \* 资源管理器（Resource Manager）提供 非代码资源的访问，如本地字符串，图形，和布局文件（ layout files ）。  
    \* 通知管理器 （Notification Manager） 使得应用程序可以在状态栏中显示自定义的提示信息。  
    \* 活动管理器（ Activity Manager） 用来管理应用程序生命周期并提供常用的导航回退功能。

有关更多的细节和怎样从头写一个应用程序，请参考 如何编写一个 Android 应用程序.

3.系统运行库

1)程序库

Android 包含一些C/C++库，这些库能被Android系统中不同的组件使用。它们通过 Android 应用程序框架为开发者提供服务。以下是一些核心库：

    \* Bionic系统 C 库 - 一个从 BSD 继承来的标准 C 系统函数库（ libc ）， 它是专门为基于 embedded linux 的设备定制的。  
    \* 媒体库 - 基于 PacketVideo OpenCORE；该库支持多种常用的音频、视频格式回放和录制，同时支持静态图像文件。编码格式包括MPEG4, H.264, MP3, AAC, AMR, JPG, PNG 。  
    \* Surface Manager - 对显示子系统的管理，并且为多个应用程序提 供了2D和3D图层的无缝融合。这部分代码  
    \* Webkit,LibWebCore - 一个最新的web浏览器引擎用，支持Android浏览器和一个可嵌入的web视图。鼎鼎大名的 Apple Safari背后的引擎就是Webkit  
    \* SGL - 底层的2D图形引擎  
    \* 3D libraries - 基于OpenGL ES 1.0 APIs实现；该库可以使用硬件 3D加速（如果可用）或者使用高度优化的3D软加速。  
    \* FreeType -位图（bitmap）和矢量（vector）字体显示。  
    \* SQLite - 一个对于所有应用程序可用，功能强劲的轻型关系型数据库引擎。

    \*还有部分上面没有显示出来的就是硬件抽象层。其实Android并非讲所有的设备驱动都放在linux内核里面，而是实现在userspace空间，这么做的主要原因是GPL协议，Linux是遵循该协议来发布的，也就意味着对 linux内核的任何修改，都必须发布其源代码。而现在这么做就可以避开而无需发布其源代码，毕竟它是用来赚钱的。而在linux内核中 为这些userspace驱动代码开一个后门，就可以让本来userspace驱动不可以直接控制的硬件可以被访问。而只需要公布这个后门代码即可。一般 情况下如果要将Android移植到其他硬件去运行，只需要实现这部分代码即可。包括：显示器驱动，声音，相机，GPS,GSM等等

2)Android 运行库

Android 包括了一个核心库，该核心库提供了JAVA编程语言核心库的大多数功能。

每一个Android应用程序都在它自己的进程中运行，都拥有一个独立的Dalvik虚拟 机实例。Dalvik被设计成一个设备可以同时高效地运行多个虚拟系统。 Dalvik虚拟机执行（.dex）的Dalvik可执行文件，该格式文件针对小内存使用做了 优化。同时虚拟机是基于寄存器的，所有的类都经由JAVA编译器编译，然后通过SDK中 的 "dx" 工具转化成.dex格式由虚拟机执行。

Dalvik虚拟机依赖于linux内核的一些功能，比如线程机制和底层内存管理机制。

4.Linux 内核

Android 的核心系统服务依赖于 Linux 2.6 内核，如安全性，内存管理，进程管理， 网络协议栈和驱动模型。 Linux 内核也同时作为硬件和软件栈之间的抽象层。其外还对其做了部分修改，主要涉及两部分修改：

1).Binder (IPC)：提供有效的进程间通信，虽然linux内核本身已经提供了这些功能，但Android系统很多服务都需要用到该功能，为了某种原因其实现了自己的一套。

2).电源管理：主要是为了省电，毕竟是手持设备嘛，低耗电才是我们的追求。

最后在谈谈Android所采用的语言，其应用开发采用java语言，我们所说的java一般包含三个部分：

1)java语言：即其语法，其写代码的程式

2).java虚拟机：为了实现一次编译到处可以运行的原则，java在编译连接以后并没有产生目标机器语言，而是采用了Java bytecode这种Java共用指令，这时就需要一个虚拟机来执行改指令。

3).库：跟我们常用的C语言一样提供一些常用的库

后两者结合就是Java Runtime Environment。

Android使用的虚拟机叫 Dalvik，最初并是不为Java设计的，它并不能运行Java bytecode指令，而是运行叫Dalvik executable，简称dx。为此Android提供了dx工具，用来将Java bytecode转换为dx。

**Android源代码结构:**

Google提供的Android包含了原始Android的目标机代码，主机编译工具、仿真环境，代码包经过解压缩后，第一级别的目录和文件如下所示：  
.  
|-- Makefile        （全局的Makefile）  
|-- bionic          （Bionic含义为仿生，这里面是一些基础的库的源代码）  
|-- bootloader      （引导加载器）  
|-- build           （build目录中的内容不是目标所用的代码，而是编译和配置所需要的脚本和工具）  
|-- dalvik          （JAVA虚拟机）  
|-- development     （程序开发所需要的模板和工具）  
|-- external        （目标机器使用的一些库）  
|-- frameworks      （应用程序的框架层）  
|-- hardware        （与硬件相关的库）  
|-- kernel          （Linux2.6的源代码）  
|-- packages        （Android的各种应用程序）  
|-- prebuilt        （Android在各种平台下编译的预置脚本）  
|-- recovery        （与目标的恢复功能相关）  
`-- system          （Android的底层的一些库）  
  
  
  
  
bionic目录展开一个级别的目录如下所示：  
bionic/  
|-- Android.mk  
|-- libc  
|-- libdl  
|-- libm  
|-- libstdc++  
|-- libthread\_db  
`-- linker  
  
  
  
bootloader目录展开的两个级别目录：  
bootloader/  
`-- legacy  
    |-- Android.mk  
    |-- README  
    |-- arch\_armv6  
    |-- arch\_msm7k  
    |-- fastboot\_protocol.txt  
    |-- include  
    |-- libboot  
    |-- libc  
    |-- nandwrite  
    `-- usbloader  
  
  
  
build目录展开的一个级别的目录如下所示：  
build/  
|-- buildspec.mk.default  
|-- cleanspec.mk  
|-- core                       （各种以mk为结尾的文件，它门是编译所需要的Makefile）  
|-- envsetup.sh  
|-- libs  
|-- target                     （包含board和product两个目录，为目标所需要文件）  
`-- tools                      （编译过程中主机所需要的工具，一些需要经过编译生成）  
  
  
  
其中，core中的Makefile是整个Android编译所需要的真正的Makefile，它被顶层目录的Makefile引用。  
envsetup.sh是一个在使用仿真器运行的时候，用于设置环境的脚本。  
dalvik目录用于提供Android JAVA应用程序运行的基础————JAVA虚拟机。  
development目录展开的一个级别的目录如下所示：  
development  
|-- apps                       （Android应用程序的模板）  
|-- build                      （编译脚本模板）  
|-- cmds  
|-- data  
|-- docs  
|-- emulator                   （仿真相关）  
|-- host                       （包含windows平台的一些工具）  
|-- ide  
|-- pdk  
|-- samples                    （一些示例程序）  
|-- simulator                  （大多是目标机器的一些工具）  
`-- tools  
  
  
  
在emulator目录中qemud是使用QEMU仿真时目标机器运行的后台程序，skins是仿真时手机的界面。  
samples中包含了很多Android简单工程，这些工程为开发者学习开发Android程序提供了很大便利，可以作为模板使用。  
external目录展开的一个级别的目录如下所示：  
external/  
|-- aes  
|-- apache-http  
|-- bluez  
|-- clearsilver  
|-- dbus  
|-- dhcpcd  
|-- dropbear  
|-- elfcopy  
|-- elfutils  
|-- emma  
|-- esd  
|-- expat  
|-- fdlibm  
|-- freetype  
|-- gdata  
|-- giflib  
|-- googleclient  
|-- icu4c  
|-- iptables  
|-- jdiff  
|-- jhead  
|-- jpeg  
|-- libffi  
|-- libpcap  
|-- libpng  
|-- libxml2  
|-- netcat  
|-- netperf  
|-- neven  
|-- opencore  
|-- openssl  
|-- oprofile  
|-- ping  
|-- ppp  
|-- protobuf  
|-- qemu  
|-- safe-iop  
|-- skia  
|-- sonivox  
|-- sqlite  
|-- srec  
|-- strace  
|-- tagsoup  
|-- tcpdump  
|-- tinyxml  
|-- tremor  
|-- webkit  
|-- wpa\_supplicant  
|-- yaffs2  
`-- zlib  
  
  
  
在external中，每个目录表示Android目标系统中的一个模块，可能有一个或者若干个库构成。其中：  
opencore为PV（PacketVideo），它是Android多媒体框架的核心。  
webkit是Android网络浏览器的核心。  
sqlite是Android数据库系统的核心。  
openssl是Secure Socket Layer，一个网络协议层，用于为数据通讯提供安全支持。  
frameworks目录展开的一个级别的目录如下所示：  
frameworks/  
|-- base  
|-- opt  
`-- policies  
  
  
  
frameworks是Android应用程序的框架。  
hardware是一些与硬件相关的库  
kernel是Linux2.6的源代码  
  
packages目录展开的两个级别的目录如下所示：  
packages/  
|-- apps  
|   |-- AlarmClock  
|   |-- Browser  
|   |-- Calculator  
|   |-- Calendar  
|   |-- Camera  
|   |-- Contacts  
|   |-- Email  
|   |-- GoogleSearch  
|   |-- HTMLViewer  
|   |-- IM  
|   |-- Launcher  
|   |-- Mms  
|   |-- Music  
|   |-- PackageInstaller  
|   |-- Phone  
|   |-- Settings  
|   |-- SoundRecorder  
|   |-- Stk  
|   |-- Sync  
|   |-- Updater  
|   `-- VoiceDialer  
`-- providers  
    |-- CalendarProvider  
    |-- ContactsProvider  
    |-- DownloadProvider  
    |-- DrmProvider  
    |-- GoogleContactsProvider  
    |-- GoogleSubscribedFeedsProvider  
    |-- ImProvider  
    |-- MediaProvider  
    `-- TelephonyProvider  
  
  
  
packages中包含两个目录，其中apps中是Android中的各种应用程序，providers是一些内容提供者（在Android中的一个数据源）。  
packages中两个目录的内容大都是使用JAVA编写的程序，各个文件夹的层次结构是类似的。  
prebuilt目录展开的一个级别的目录如下所示：  
prebuilt/  
|-- Android.mk  
|-- android-arm  
|-- common  
|-- darwin-x86  
|-- linux-x86  
`-- windows  
  
  
  
system目录展开的两个级别的目录如下所示：  
system/  
|-- bluetooth  
|   |-- bluedroid  
|   `-- brfpatch  
|-- core  
|   |-- Android.mk  
|   |-- README  
|   |-- adb  
|   |-- cpio  
|   |-- debuggerd  
|   |-- fastboot  
|   |-- include            （各个库接口的头文件）  
|   |-- init  
|   |-- libctest  
|   |-- libcutils  
|   |-- liblog  
|   |-- libmincrypt  
|   |-- libnetutils  
|   |-- libpixelflinger  
|   |-- libzipfile  
|   |-- logcat  
|   |-- logwrapper  
|   |-- mkbootimg  
|   |-- mountd  
|   |-- netcfg  
|   |-- rootdir  
|   |-- sh  
|   `-- toolbox  
|-- extras  
|   |-- Android.mk  
|   |-- latencytop  
|   |-- libpagemap  
|   |-- librank  
|   |-- procmem  
|   |-- procrank  
|   |-- showmap  
|   |-- showslab  
|   |-- sound  
|   |-- su  
|   |-- tests  
|   `-- timeinfo  
`-- wlan  
    `-- ti

1、android文件系统的结构  
android源码编译后得到system.img,ramdisk.img,userdata.img映像文件。其中， ramdisk.img是emulator的文件系统，system.img包括了主要的包、库等文件，userdata.img包括了一些用户数 据，emulator加载这3个映像文件后，会把 system和 userdata分别加载到 ramdisk文件系统中的system和 userdata目录下。因此，我们可以把ramdisk.img里的所有文件复制出来，system.img和userdata.img分别解压到 ramdisk文件系统中的system和 userdata目录下。  
2、分离android文件系统出来  
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：  
file ramdisk.img，输出：   
ramdisk.img: gzip compressed data, from Unix, last modified: Wed Mar 18 17:16:10 2009   
Android源码编译后除了生成system.img，userdata.img之外还生成system和 userdata文件夹，因此不需要解压它们。Android源码编译后还生成root文件夹，其实root下的文件与 ramdisk.img 里的文件是一样的，不过这里还是介绍怎样把 ramdisk.img解压出来:  
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   
gunzip ramdisk.img.gz   
然后新建一个文件夹，叫ramdisk吧，进入，输入命令   
cpio -i -F ../ramdisk.img   
这下，你就能看见并操作ramdisk里面的内容了。   
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。这样就得到一个文件系统了。  
3、使用网络文件系统方式挂载android文件系统  
因此，我们需要建立/nfsroot目录，再建立/nfsroot/androidfs目录，把刚才的android文件系统改名为androidfs，并链接到/nfsroot/androidfs  
4、android内核引导文件系统  
android内核挂载/nfsroot/androidfs之后，根据init.rc,init.goldfish.rc来初始化并装载系统库、程序等直到开机完成。init.rc脚本包括了文件系统初始化、装载的许多过程。init.rc的工作主要是：  
1）设置一些环境变量  
2）创建system、sdcard、data、cache等目录  
3）把一些文件系统mount到一些目录去，如，mount tmpfs tmpfs /sqlite\_stmt\_journals  
4）设置一些文件的用户群组、权限  
5）设置一些线程参数  
6）设置TCP缓存大小