

# Android系统定制

## Fork一个新的产品版本

**Android**PlatformDeveloper'sKit简称PDK,是Android为厂商设计用于快速版本升级.

(Android 定制过程,官方详解<http://www.kandroid.org/online-pdk/guide/index.html>)

1Android 版本定义的层级:

1) Product

myProduct,myProduct\_eu,myProduct\_eu\_fr,j2,sdk

Theproductlayerdefinesacompletespecificationofashippingproduct,definingwhichmodulestobuildandhowtoconfigurethem.Youmightofferadeviceinseveraldifferentversionsbasedonlocale,forexample,oronfeaturessuchasacamera.

2) Device

myDevice,myDevice\_eu,myDevice\_eu\_lite

Thedevicelayerrepresentsthephysicallayerofplasticonthedevice.Forexample,NorthernAmericandevicesprobablyincludeQWERTYkeyboardswhereasdevicessoldinFranceprobablyincludeAZERTYkeyboards.Peripheralstypicallyconnecttothedevicelayer.

3) Board

sardine,trout,goldfish

The board layer represents the bare schematics of a product. You may still connect peripherals to the board layer.

#### 4) Arch

arm(arm5te)(arm6),x86,68k

The arch layer describes the processor running on your board.

## 2 添加一个新的 product

### 1) Create product directory:

```
cd android2.3.4_32bit
```

```
mkdir vendor/<company_name>
```

```
mkdir vendor/<company_name>/products/
```

2) Create a product-specific makefile, called vendor/<company\_name>/products/<first\_product\_name>.mk

此 Makefile 可能参数参考 [http://www.kandroid.org/online-pdk/guide/build\\_new\\_device.html#androidBuildSystemProductDefFiles](http://www.kandroid.org/online-pdk/guide/build_new_device.html#androidBuildSystemProductDefFiles)

和 vendor/forlinx/products/OK6410.mk

OK6410 里面定义了需要安装的 APK, 去掉了 Email 等 APK.

### 3) create an AndroidProducts.mk

指向刚创建的<first\_product\_name>.mk.

#### 4)退到上级

mkdir vendor/<company\_name>/<board\_name>

createBoardConfig.mk

createsystem.proptomodifysystemsetting,thisoverridesettingsintheproducts/  
generic/system.prop/build/target/board/generic/system.propfile

#### 5)测试

.build/envsetup.sh

makePRODUCT-<first\_product\_name>-user

OK6410 使用的变通的作法,直接导出一些设定值.

#### 6)最终结构

<company\_name>

<board\_name>

Android.mk

product\_config.mk

system.prop

products

AndroidProducts.mk

<first\_product\_name>.mk

<second\_product\_name>.mk

Fork 之后,可以参考官网的说明,逐步修改.编译的元子单位是 product,所以不同的厂商,硬件的产品名称必须是不同的.这部分主要是一些系统配置,非开源固件的释放等.配置后新编译出来的系统的版本信息也会跟着改变.

## 定制启动画面

### 设置Linux内核启动LOGO

#### 1、让内核支持启动 LOGO

在内核目录中 makemenuconfig

DeviceDrivers---->Graphicssupport---->

选中 Bootuplogo---->Standard224-colorLinuxlogo

#### 2、制作开机 logo 的图片

将要当作开机 LOGO 的图片按照你屏幕的大小进行裁减,并将其保存为 bmp 格式或 png 格

式;例如 linuxlogo.bmp 或 linuxlogo.png

在 linux 下输入以下命令(forbmp):

```
#bmptoppmlinuxlogo.bmp>linuxlogo.ppm //生成 ppm
```

```
#ppmquant224linuxlogo.ppm>linuxlogo224.ppm//转换成 224 颜色
```

```
#pnmnorawlinuxlogo224.ppm>linuxlogo224.ppm//转换成 ascii 格式
```

在 Linux 下输入以下命令(forpng) :

```
#pngtopnmlinuxlogo.png>linuxlogo.pnm
```

```
#pnmquant224linuxlogo.pnm>linuxlogo224.pnm
```

```
#pnmtoplainpnm linuxlogo224.pnm>linuxlogo224.ppm
```

注：上述命令需要安装 netpbm 的工具包 `sudo apt-get install netpbm`

3、将 LOGO 文件放入内核中

拷贝到制作好的 LOGO 文件到 `/drivers/video/logo` 文件夹

替换 `logo_linux_clut224.ppm` 文件，还要删除该文件夹下的 `logo_linux_clut224.o`

4、重新编译内核 `make zImage` 或 `make uImage`

## 修改android开机画面

Android 系统开机动画包括两部分：

1 开机显示的 ANDROID 文字;

2 ANDROID 发光动画。

### 一 开机显示的 ANDROID 文字

1、制作 `initlogo.rle`

文件位于 `vender/hdsy/HDSY/initlogo.rle`.生成方法:

使用 PS 制作一张 320\*480 的图片，保存时选“保存为 Web 所用格式”，然后在弹出的窗口上，“预设”项选择“PNG-24”，保存为 `android_logo.png`

2、将图片转换为 raw 格式

使用 linux 下的 ImageMagick 自带的 `convert` 命令，进行 raw 格式转换，命令为：

```
convert android_logo.png  rgb:android_logo.raw
```

注：ubuntu10.04 默认已经安装 ImageMagick 工具，如果当前系统没有安装，可以执行下面的命令安装：

```
sudo apt-get install imagemagick
```

### 3、将 raw 格式转化为 rle 文件

需要用到 android 编译后的 rgb2565 工具，在 android/out/host/linux-x86/bin 目录下(android 为当前源码所在目录)，转换命令如下：

```
rgb2565 -rle <android_logo.raw> initlogo.rle
```

也可以使用下载的合适尺寸的 jpg 图片制作,我写了一个脚本辅助转换:

```
#!/bin/bash
for file in `ls *.jpg *.jpeg`
do
    echo $file
    convert $file rgb:$file.r
    rgb2565 -rle <$file.r> $file.rle
done
```

到目前为止，启动需要显示的图像已经做好了，就是 initlogo.rle，替换原来的 initlogo.rle 文件即可。

### 4.相关的实现代码，摘自 Android2.1

```
//启动时载入图像，如果没有找 INIT_IMAGE_FILE 到这个文件名的图片，
//则打印 ANDROID 字样。

if(load_565rle_image(INIT_IMAGE_FILE)){
    fd=open("/dev/tty0",O_WRONLY);
    if(fd>=0){
        constchar*msg;
        msg="\n"
        "\n"
        "\n"
        "\n"
        "\n"
        "\n"
        "\n//consoleis40colsx30lines
        "\n"
        "\n"
```

```

"\n"

"\n"

"\n"

"\n"

"ANDROID";

write(fd,msg,strlen(msg));

close(fd);

}

}

// "system/core/init/init.c"1039 行--86%--

```

## 5.图像文件名的指定

```

//图像文件名宏的定义 INIT_IMAGE_FILE

void queue_all_property_triggers();

#define INIT_IMAGE_FILE "/initlogo.rle"

int load_565rle_image(char*file_name);

// "system/core/init/init.h"179 行--100%--

```

## 二 开机动画(闪动的 ANDROID 字样的动画图片)

Android 的系统登录动画类似于 Windows 系统的滚动条，是由前景和背景两张 PNG 图片组成,一个是前景图片

frameworks/base/core/res/assets/images/android-logo-mask.png

文字 logo 部分镂空,另一个是背景图片 ( android-logo-shine.png ),是简单的纹理.

主要修改第一个文件.直接用 gimp 生成一个等大的 png 图片替换即可.

系统登录时，前景图片在最上层显示，程序代码控制背景图片连续滚动，透过前景图片文字镂空部分滚动显示背景纹理，从而实现动画效果。

相关代码：

/frameworks/base/cmds/bootanimation/BootAnimation.h

/frameworks/base/cmds/bootanimation/BootAnimation.cpp

/frameworks/base/core/res/assets/images/android-logo-mask.png

Android 默认的前景图片，文字部分镂空，大小 256×64

/frameworks/base/core/res/assets/images/android-logo-shine.png

Android 默认的背景图片，有动感效果，大小 512×64

## 定制launcher

Android Launcher 位于 packages/apps/Launcher2

去网上下载一个开源Launcher替换(CyanogenMod <http://zh.wikipedia.org/wiki/CyanogenMod>. 目前最新的 CyanogenMod 版本是基于Android 4.1 下去开发，并加入大量自制的源代码。CyanogenMod 的源代码本来是由 Cyanogen (本名Steve Kondik) 编写，后来也加入了[xda-developers](#)社区强化过的主画面、调用程序和[网页浏览器](#)，另外也加入 [BusyBox](#)等其它的程序。):

[https://github.com/CyanogenMod/android\\_packages\\_apps\\_ADWLauncher](https://github.com/CyanogenMod/android_packages_apps_ADWLauncher)

下载源码:

git clone [https://github.com/CyanogenMod/android\\_packages\\_apps\\_ADWLauncher](https://github.com/CyanogenMod/android_packages_apps_ADWLauncher)  
[er.git](#)

添加到系统编译树中:

mv [https://github.com/CyanogenMod/android\\_packages\\_apps\\_ADWLauncher](https://github.com/CyanogenMod/android_packages_apps_ADWLauncher) packages/apps/ ADWLauncher



编译新 Launcher

```
mmm packages/apps/ ADWLauncher
```

替换系统原来的 Launcher:

```
adb install -r out/target/product/HDSY/system/app/ADWLauncher.ap
```

k

如果需要修改,可以用 eclipse 打开这个 Launcher 进行编辑.