

zhenwenxian的专栏

RSS订阅

个人资料



zhenwenxian

关注

原创	粉丝	喜欢	评论
270	661	7	160

等级：

博客 7

访问：189万+

积分：1万+

排名：740

最新文章

ubuntu 12.04（64位）下搭建android5.0开发环境

基于短消息的远程家电红外遥控系统

步进电机原理和驱动

android 流量的统计

驱动程序的健壮性考虑

个人分类

DPPM	1篇
ssh	2篇
Ubuntu的root帐号	2篇
sudo passwd root	1篇
HDQ 单线通信协议 时序解析 Bq27541	1篇

展开

归档

2014年12月	1篇
2014年8月	1篇
2014年6月	1篇


7


收藏


评论


微信


微博


QQ

展开

热门文章

- 将tgz文件解压到指定目录

阅读量：120046
- 锂电池充电的原理

阅读量：69559
- 普通充电器给苹果IPHONE/IPAD2充电的USB端的识别电阻的设置

阅读量：56156
- android的m、mm、mmm编译命令的使用

阅读量：51581
- 如何解包 / 编辑 / 打包boot.img文件

阅读量：47181

最新评论

- 工作队列的使用例子

xiaochechuan4436：很通俗易懂
- P沟道mos管作为开关的条件（GS...

u013135755：图呢 sb
- android 流量的统计

rza1314：[reply]chenxiang890416[/reply] 因为博客是网上抄过来的，自己也不知道呀
- 如何制作LINUX的patch文件...

qq_38625051：可以，通俗易懂，比其他人写的干巴巴的东西好多了
- 防反接保护电路

qq_39392944：C1和R3是干嘛的

联系我们




请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务  百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

原 android boot.img 结构

2011年03月02日 23:15:00

阅读量：23022

android 的boot.img 包括 boot header , kernel , ramdisk

首先来看看Makefile是如何产生我们的boot.img的：

boot镜像不是普通意义上的文件系统，而是一种特殊的Android定制格式，由文件头信息boot header，压缩的内核，文件系统数据ramdisk以及second stage load

文件头信息的具体结构可以在system/core/mkbootimg/bootimg.h中看到：

```
struct boot_img_hdr
{
    unsigned char magic[BOOT_MAGIC_SIZE];
    unsigned kernel_size;
    unsigned kernel_addr;
    unsigned ramdisk_size;
    unsigned ramdisk_addr;
    unsigned second_size;
    unsigned second_addr;
    unsigned tags_addr;
    unsigned page_size;
    unsigned unused[2];
    unsigned char name[BOOT_NAME_SIZE]
    unsigned char cmdline[BOOT_ARGS_SIZE]
    unsigned id[8]; //存放时间戳，校验和，SHA加密等内容
}
```

boot.img文件跳过4k的文件头之后，包括两个 gz包，一个是boot.img-kernel.gz：Linux内核，一个是boot.img-ramdisk.cpio.gz

大概的组成结构如下

```
*
** +-----+
** | boot header | 1 page
** +-----+
** | kernel      | n pages
** +-----+
** | ramdisk     | m pages
** +-----+
** | second stage | o pages
** +-----+
```

boot header为包括命令行参数等等,地址为000-----0xFFF

ramdisk为 1F8B080000000000开头

kernel为 0000A0E1 重复8遍开头

关于boot header这个数据结构我们需要重点关注，在这里我们关注其中几个比较重要的值，这些值定义在boot/boardconfig.h里面，不同的芯片对应vendor下不同的boardconfig，在这里我们的值分别是（分别是kernel/ramdisk/tags载入ram的物理地址）：

```
#define PHYSICAL_DRAM_BASE 0x00200000
#define KERNEL_ADDR      (PHYSICAL_DRAM_BASE + 0x00008000)
#define RAMDISK_ADDR      (PHYSICAL_DRAM_BASE + 0x01000000)
#define TAGS_ADDR         (PHYSICAL_DRAM_BASE + 0x00000100)
#define NEWTAGS_ADDR      (PHYSICAL_DRAM_BASE + 0x00004000)
```

上面这些值分别和我们开篇时候提到的那几个名词相对应，比如kernel_addr就是ZTEXTADDR，RAMDISK_ADDR就是INITRD_PHYS,而TAGS_ADDR就是PARAMS_PHYS。bootloader会从boot.img的分区中将kernel和ramdisk分别读入RAM上面定义的地址中，然后就会跳到ZTEXTADDR开始执行。

```

./init.trout.rc
./default.prop
./proc
./dev
./init.rc
./init
./sys
./init.goldfish.rc
./sbin
./sbin/adbd
./system
./data

```

如果要分离可以用winhex将boot.img打开

找到0000A0E1 到1F8B0800000000的前面的数据块保持为kernel

找到1F8B0800000000到文件尾部的数据块保持为ramdisk.img

```

out/host/linux-x86/bin/mkbootimg --kernel out/target/product/msm7630_surf/kernel --ramdisk out/target/product/msm7630_surf/ramdisk.img --cmdline "console=ttyMSM1,115200n8 androidboot.hardware=qcom" --base 0x00200000 --pagesize 4096 --output out/target/product/msm7630_surf/boot.img

```

根据上面的命令我们可以首先看看mkbootimg 这个工具的源文件：system/core/mkbootimg.c。看完之后我们就能很清晰地看到boot.img的内部构造，它是由boot header /kernel /ramdisk /second stage构成的，其中前3项是必须的，最后一项是可选的。mkbootimg分析参数后，依次写入header, kernel ,ramdisk .

```

header + padding + kernel + padding + ramdisk + padding + ...
4 * 2, magic, 固定为"ANDROID!"
4 * 1, kernel长度, 小端unsigned
4 * 1, kernel地址, 应为base + 0x00008000 (base为0x200000)
4 * 1, ramdisk长度, 小端unsigned
4 * 1, ramdisk地址, 应为base + 0x01000000
4 * 1, second stage长度, 小端unsigned, 为0
4 * 1, second stage地址, 应为base + 0x00f00000
4 * 1, tags地址, 应为base + 0x00000100
4 * 1, page大小, 小端unsigned, 为2048或者4096

4 * 2, 未使用, 固定为0x00
4 * 4, 板子名字, 一般为空
4 * 128, 内核命令参数, 为mem=211M console=ttyMSM2,115200n8 androidboot.hardware=qcom console=ttyUSB_CONSOLE0 androidboot.console=ttyUSB_CONSOLE0
4 * 8, id, 为sha之类, 实际写0x00就可
padding, 以上header为608字节, 把这部分补齐到page_size * 2大小
kernel_size, kernel内容
padding, 把kernel_size补齐到page_size * 2
ramdisk_size, ramdisk内容
padding, 把ramdisk补齐到page_size * 2
second_size, second内容, 一般为0

```

配合 boot.img 来看会比较好理解.

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF
000000	14E	4452	4F49	4421	0855	3800	0080	2040	ANDROID!.U8..@
000010	60A5	0200	0000	2041	0000	0000	0000	1041	Y.... A.....A
000020	0001	2040	0008	0000	0000	0000	0000	0000	.. @.....
000030	0000	0000	0000	0000	0000	0000	0000	0000
000040	766D	616C	6C6F	633D	3236	344D	2063	6F6E	vmalloc=264M con
000050	736F	6C65	3D74	7479	4853	4C30	2C31	3135	sole=ttyHSL0,115
000060	3230	302C	6E38	2061	6E64	726F	6964	626F	200,n8 androidbo
000070	6F74	2E68	6172	6477	6172	653D	7163	6F6D	ot.hardware=qcom

由此可知 boot_img_hdr 中各成员值为：

```

magic[BOOT_MAGIC_SIZE] = "ANDROID!"
kernel_size             = 0x00385508
kernel_addr             = 0x40208000
ramdisk_size            = 0x0002A560
ramdisk_addr            = 0x41200000
second_size             = 0x00000000
second_addr             = 0x41100000
tags_addr               = 0x40200100
page_size               = 0x00000800

cmdline[BOOT_ARGS_SIZE] = "vmalloc=264M console=ttyHSL0,115200,n8 androidboot.hardware=qcom"

```

```

MEMBASE := 0x40100000 # SMI
MEMSIZE := 0x00100000 # 1MB

BASE_ADDR      := 0x40200000
TAGS_ADDR      := BASE_ADDR+0x00000100
KERNEL_ADDR    := BASE_ADDR+0x00008000
RAMDISK_ADDR   := BASE_ADDR+0x01000000
SCRATCH_ADDR   := 0x48000000

```

TAGS_ADDR 如上 target/<your-platform>/rules.mk 所定义的：0x40200100, 所以 boot_linux(), 就是传入TAGS_ADDR,

然后将资料写入 tag, tag 的结构如下所示.

```

struct boot_img_hdr
{
    unsigned char magic[BOOT_MAGIC_SIZE];

    unsigned kernel_size; /* size in bytes */
    unsigned kernel_addr; /* physical load addr */

    unsigned ramdisk_size; /* size in bytes */
    unsigned ramdisk_addr; /* physical load addr */

    unsigned second_size; /* size in bytes */
    unsigned second_addr; /* physical load addr */

    unsigned tags_addr; /* physical addr for kernel tags */
    unsigned page_size; /* flash page size we assume */
    unsigned unused[2]; /* future expansion: should be 0 */

    unsigned char name[BOOT_NAME_SIZE]; /* asciiz product name */



    unsigned char cmdline[BOOT_ARGS_SIZE];
}

```

然后进入到 kernel 的入口函数: entry(0, machtype, tags)

文章标签： android header tags linux内核 makefile


想对作者说点什么？ 我来说两句

- **mobz** 2014-07-24 10:04:38 #2楼
先收藏了留着以后验证下。
- **evilcode** 2011-07-07 18:57:34 #1楼
请教下boot.img 打包的是ulmage还是zImage,是ramdisk还是uramdisk？看你写的文件开头的部分应该好似zImage和ramdisk，如果我使用uboot引导的话，那么需要使用ulmage和uramdisk？OTA升级里面有把boot.img分成内核和文件系统吗？貌似是直接全烧进去某一个分区的？

上一页 1 下一页

怎样修改安卓bootimg内核

Android 产品中，内核格式是Linux标准的zImage，根文件系统采用ramdisk格式。这两者在Android下是直接合并在一起取名为boot.img,会放在一个独立分区当中。这个分区格式是...

 bfboys 2016-09-17 11:19:04 阅读数：1984

[Android]构建boot.img

[Android]构建boot.img(一):root目录与ramdisk.img的生成 以TCC88XX为例，当在Android顶层源码目录使用make编译完成后，会生成这样一个目录：ou...

 wzw88486969 2013-07-07 20:15:58 阅读数：4846


<为知更新>制作 ramdisk.img，使用cpio 和 gzip

linux2.6 内核支持两种格式的 initrd（虚拟文件系统），一种是 linux2.4 内核那种传统格式的文件系统镜像 image-initrd，其核心文件就是 /linuxrc. 另外一种格式...

 wh_19910525 2012-10-25 19:29:30 阅读数：10885

android 启动与各镜像文件的关系

Android启动过程 Android在启动的时候，会由UBOOT传入一个init参数，这个init参数指定了开机的时候第一个运行的程序，默认就是init程序，这个程序在ramdisk...

 prike 2016-08-08 09:22:39 阅读数：711


Linux系统下ramdisk文件解压缩与压缩处理

Linux系统下ramdisk文件解压缩与压缩处理

 hunter168_wang 2016-10-27 11:28:42 阅读数：1336

读取boot.img头（根据Android源码中的bootimg.h读取）将kernel和ramdisk读取出来

使用bootimg.pl（linux、win7都可以执行）脚本，该脚本可以读取boot.img头（根据Android源码中的bootimg.h读取）将kernel和ramdisk读取出来，此脚本也会输...


 JINCHENG121 2012-12-08 16:41:43 阅读数：2035

android rom制作之bootimg的详细介绍和使用

 wyw594 2013-11-20 18:07:35 阅读数：1394

Android boot.img的由来

编译完android源码后，以飞思卡尔的IMX6Q为例，生成的镜像在目录out/target/product/sabresd_6dq中 这些镜像包括：boot.img kernel ra...

 runfan1014 2017-02-08 17:56:46 阅读数：1419

android解析 ramdisk.img boot.img system.img

img解析: ramdisk.img:android根文件系统,在android编译系统生成的out/target/product/root目录中 结构: ./init.trout...

 wzw88486969 2013-07-17 09:28:22 阅读数：6505

制作img镜像文件的5种方法

1. 在DOS下用debug 把floppy.img写入A盘 debug floppy.img -w 100 0 0 1 -q 把floppy.img写入B盘 debu...

 yunkai666 2012-10-22 01:36:46 阅读数：8291

Android编译过程总结及android中各种img文件的作用以及系统启动过程

1、编译环境的准备 (1) 确保安装有ubuntu系统或者虚拟机 (2) 安装JDK1.6 (对于Android2.3以上代码) \$ sudo add-apt-repository "debhttp...

 baiyongtask 2015-01-26 17:10:43 阅读数：2562

u-boot镜像Image中有关结构体

(uboot1.1.6为分析对象) 在uboot启动阶段 do_bootm_linux往往会分析内核镜像，这里面会有几个结构体 image_header_t 里面定义了镜像的头部 typedef st...

 comwise 2013-10-19 11:16:13 阅读数：1733

签名boot.img及system.img和verity_key的生成

请参考 <http://luomingmao.com/2016/08/29/Verified-Boot/>

 czq7511 2017-02-21 15:18:21 阅读数：1995

boot.img的解包与打包

Android 产品中，内核格式是Linux标准的zImage，根文件系统采用ramdisk格式。这两者在Android下是直接合并在一起取名为boot.img,会放在一个独立分区当中。这个分区格式是...

 wh_19910525 2012-11-19 17:27:39 阅读数：66013

解包、编辑、打包boot.img文件

首先声明这是转帖，Linux环境大家可以用VMWARE来虚拟，可以下载UBUNTU 目录 1、背景知识 2、boot和recovery映像的文件结构3、对映像文件进行解包、编辑、打包的常规方...

 zhangmiaoping23 2016-12-25 00:13:53 阅读数：3014

boot.img和recovery.img结构说明

今天对boot.img和recovery.img结构做了研究，这是一个十分好玩的事情，当然，在android移植和编译的过程中，也是需要有一定了解的。 Why：其实为什么要做这个了解呢，起...

 g1im2 2015-05-15 15:42:12 阅读数：2174

关于boot.img和recovery.img的修改和编辑

关于boot.img和recovery.img的编辑和修改方面的文章，希望能够为感兴趣的朋友节约一些看资料的时间。感谢本文的作者：Alansj, DarkrifX, RyeBrye, Will, T...

 u011467537 2016-11-21 19:44:54 阅读数：4083

使用fastboot工具刷入 recovery.img boot.img system.img等

 Kitty_Landon 2017-01-04 10:52:53 阅读数 : 22747

安卓**Android** ROM定制、移植，安卓软件反编译、汉化实战教程第六篇：**boot.img**、**recovery.img**的解包、打包！

太抱歉了，因为教程实在是有点粗浅了，其实很多东西都不知道怎么去写，这不是复制粘贴，当然很多只是一步步的走的，不过也许我自己觉得已经阐述的很清楚了，可是一旦别人看起来，还是感觉很深奥，没办法，本人就这点...

 sky79 2014-09-13 17:45:09 阅读数 : 1559

android打包解包**boot.img**,**system.img**

原帖地址：<http://www.52pojie.cn/thread-488025-1-1.html> 转载Mark一下，日后研究 最近工作需要对**boot.img**，**system.img**进行破解...

 u013463707 2017-04-28 15:00:36 阅读数 : 1361