一、设备和环境

设备 MSIDIGTAL-RM701

平台 瑞芯微RK3128

环境 Windows 11 Subsystem Ubuntu 20.04.5 LTS

二、工具

在打包解包的过程中，我们至少需要使用瑞芯微官方提供的三个工具：afptool、img\_unpack、img\_maker。可以自己通过源码编译，也可以使用我文章后面附上的成品，架构为linux\_amd64

#拉取源码

git clone 'https://github.com/TeeFirefly/rk2918\_tools.git'

#进入目录

cd rk2918\_tools/

#生成

make

#将工具复制到/usr/lcoal/bin以方便调用

sudo cp afptool img\_unpack img\_maker mkkrnlimg /usr/local/bin

三、解包

1.解包release\_update.img

从网上下载到的成品固件，我们称之为release\_update.img，使用工具img\_unpack将其解包为loader.img和update.img

#当前目录存放有：release\_update.img

#第一个参数指定固件，第二个参数指定保存目录

img\_unpack release\_update.img img

2.解包update.img

通过afptool -unpack把update.img解包出各个分区镜像：

#进入img目录

cd img

#当前目录内有：loader.img update.img

#使用afptool对update.img进行解包

#指定工作模式，解包为-unpack，打包为-pack；第一个参数指定镜像位置；第二个参数指定保存目录

afptool -unpack update.img update

3.目录结构

至此，解包已经完成，返回上一级目录，列出目录结构应是如此：

jamaskii@HP-445R-G6:/mnt/d/rk3128/demo/img$ cd ../

jamaskii@HP-445R-G6:/mnt/d/rk3128/demo$ tree .

.

├── img

│ ├── loader.img

│ ├── update

│ │ ├── Image

│ │ │ ├── MiniLoaderAll.bin

│ │ │ ├── boot.img

│ │ │ ├── kernel.img

│ │ │ ├── misc.img

│ │ │ ├── parameter.txt

│ │ │ ├── recovery.img

│ │ │ ├── resource.img

│ │ │ ├── system.img

│ │ │ └── uboot.img

│ │ ├── RESERVED

│ │ └── package-file

│ └── update.img

└── release\_update.img

3 directories, 14 files

四、修改

本次以给固件加入ROOT权限为例，阐述修改system分区的一些步骤。

1.镜像扩容

在修改Image/system.img的内容之前，首先需要对其进行扩容，不然在后续的操作中大概率会遇到空间不足的错误警告：

jamaskii@HP-445R-G6:/mnt/d/rk3128/demo/img/update$ sudo cp '/mnt/d/rk3128/supersu/armv7/su' '/mnt/system/xbin'

cp: error writing '/mnt/system/xbin/su': No space left on device

以对其扩容128MB为例：

#填充128MB数据

dd if=/dev/zero bs=1M count=128 >> Image/system.img

#检查镜像错误并修正、扩容

e2fsck -f Image/system.img

resize2fs Image/system.img

2.镜像挂载

解包出来的Image/system.img本质上是一个ext4文件系统的镜像，所以修改的思路是将其挂载在/mnt/system，当然其他位置也可以，然后进去对内容进行修改：

#创建挂载点

sudo mkdir /mnt/system

#进入img/update/目录

cd img/update/

#当前目录内有：Image RESERVED package-file

#将Image/system.img挂载到/mnt/system

sudo mount Image/system.img /mnt/system

3.增加su二进制文件

到SuperSU官网下载其最新的zip包，解压，查看/META-INF/com/google/android/update-binary脚本，分析需要进行哪些操作。

首先是根据判断API等级：

# binary ARCH/path build type API

#

# arm-v5te arm ndk non-pie 7+

# x86 x86 ndk non-pie 7+

#

# x86 x86 ndk pie 17+ (su.pie, naming exception)

# arm-v7a armv7 ndk pie 17+

# mips mips ndk pie 17+

#

# arm64-v8a arm64 ndk pie 20+

# mips64 mips64 ndk pie 20+

# x86\_64 x64 ndk pie 20+

RK3128属于arm-v7，所以匹配API17+

然后查看对应的操作有哪些。由于篇幅限制，这里只节选API17+部分：

# API source target chmod

#

# 7-19 common/Superuser.apk /system/app/Superuser.apk 0644

# 20+ common/Superuser.apk /system/app/SuperSU/SuperSU.apk 0644

#

# 17+ common/install-recovery.sh /system/etc/install-recovery.sh 0755

# 17+ /system/bin/install-recovery.sh 0755

#

# 7+ ARCH/su /system/xbin/su \*3

# 7+ /system/bin/.ext/.su \*3

# 17+ /system/xbin/daemonsu 0755

# \*3: 06755 if API < 18, 0755 if API >= 18

#

# 17+ common/99SuperSUDaemon \*6 /system/etc/init.d/99SuperSUDaemon 0755

# 17+ 'echo 1 >' or 'touch' \*7 /system/etc/.installed\_su\_daemon 0644

# \*6: only place this file if /system/etc/init.d is present

# \*7: the file just needs to exist

五、打包

首先需要将修改后的分区卸载：

sudo umount /mnt/system

1.打包新的update.img

#当前目录内有Image RESERVED package-file parameter

#设定工作模式为打包-pack，第一个参数为解包后的目录（包含Image/和package-file）；第二个参数为保存位置

afptool -pack . ../update\_new.img

如果打包工具抛出如下错误：

jamaskii@HP-445R-G6:/mnt/d/rk3128/demo/img/update$ afptool -pack . ../update\_new.img

------ PACKAGE ------

Can't open file: ./parameter

Pack failed

则是在当前目录内找不到parameter，我们需要将Image/parameter.txt复制到当前目录，并命名为parameter

cp Image/parameter.txt parameter

2.打包新的release\_update.img

#回到上级目录

cd ../

#此时目录内有loader.img update update.img update\_new.img

#其中update\_new.img是我们新打包出来的镜像

#设置芯片类型为rk3xx系列，第一个参数为loader.img位置；第二个参数为update.img位置；第四个参数为保存位置

img\_maker -rk31 loader.img update\_new.img ../release\_update\_new.img

说一下这里的打包工具img\_maker，看下他的帮助信息：

usage: img\_maker [chiptype] <loader> <old image> <out image>

chiptype:

-rk29

-rk30

-rk31

-rk32

-rk33

If chiptype is missing, it is default to -rk32.

一定要记得指定芯片类型，如果忘记了，则按默认的rk32xx系列打包。到刷机时如果目标设备不是rk32xx，则会验证出错。

打包完成，我们就得到了新的固件'../release\_image\_new.img'，可以用这个固件进行刷机了。

六、值得注意的点

1.区分release\_update.img和update.img

前者是直接用来刷机的固件，后者仅仅包含各个分区的镜像文件。他们的关系如图：

2.在打包release\_update.img时记得指定芯片类型

否则做出来的包可能会导致刷机失败。

七、附件

工具：[蓝奏云]rk3128\_firmware\_tools\_linux\_amd64.zip

————————————————

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

原文链接：https://blog.csdn.net/u013538358/article/details/128890082