Ubuntu16.04 麒麟系统下 Firefly-RK3399 开发环境的搭建流程及内核编译

1、进入 Ubuntu16.04 麒麟系统，打开终端界面，安装开发包：
sudo apt-get install build-essential lzop libncurses5-dev libssl-dev
# 如果使用的是 64 位的 Ubuntu，还需要安装：
sudo apt-get install libc6:i386

```
mo@mo-N55SL:~$ sudo apt-get install build-essential lzop libncurses5-dev libssl-dev
[sudo] mo 的密码：
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
build-essential 已经是最新版 (12.1ubuntu2)。
libncurses5-dev 已经是最新版 (6.0+20160213-1ubuntu1)。
lzop 已经是最新版 (1.03-3.2)。
libssl-dev 已经是最新版 (1.0.2g-1ubuntu4.6)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 336 个软件包未被升级。
mo@mo-N55SL:~$ sudo apt-get install libc6:i386
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
libc6:i386 已经是最新版 (2.23-0ubuntu7)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 336 个软件包未被升级。
mo@mo-N55SL:~$
```

2、安装 mkbootimg 工具
git clone https://github.com/neo-technologies/rockchip-mkbootimg.git
cd rockchip-mkbootimg
make && sudo make install
注：若没安装 git 包，先输入 sudo apt install git 进行安装

```
mo@mo-N55SL:~$ git clone https://github.com/neo-technologies/rockchip-mkbootimg.git
正克隆到 'rockchip-mkbootimg'...
remote: Counting objects: 143, done.
remote: Total 143 (delta 0), reused 0 (delta 0), pack-reused 143
接收对象中: 100% (143/143), 35.72 KiB | 0 bytes/s, 完成.
处理 delta 中: 100% (82/82), 完成.
检查连接... 完成。
mo@mo-N55SL:~$ cd rockchip-mkbootimg
mo@mo-N55SL:~/rockchip-mkbootimg$ make&&sudo make install
cc -O2 -Wall -Wextra -o afptool afptool.c -lcrypto
cc -O2 -Wall -Wextra -o img_maker img_maker.c -lcrypto
cc -O2 -Wall -Wextra -o mkbootimg mkbootimg.c -lcrypto
cc -O2 -Wall -Wextra -o unmkbootimg unmkbootimg.c -lcrypto
install -d -m 0755 /usr/local/bin
install -m 0755 afptool img_maker mkbootimg unmkbootimg /usr/local/bin
install -m 0755 mkrootfs mkupdate mkcpiogz unmkcpiogz /usr/local/bin
mo@mo-N55SL:~/rockchip-mkbootimg$
```

3、获取内核源码和安装交叉编译工具链
mkdir project
cd project
Firefly-RK3399 Linux 内核代码获取方式:
git clone https://TeeFirefly@gitlab.com/TeeFirefly/linux-kernel.git.
获取交叉编译工具:
git clone https://TeeFirefly@gitlab.com/TeeFirefly/prebuilts.git.

```
mo@mo-N55SL:~$ mkdir project
mo@mo-N55SL:~$ cd project
mo@mo-N55SL:~/project$ git clone https://TeeFirefly@gitlab.com/TeeFirefly/linux-kernel.git
正克隆到 'linux-kernel'...
remote: Counting objects: 4770015, done.
remote: Compressing objects: 100% (2584/2584), done.
remote: Total 4770015 (delta 4229), reused 5023 (delta 3701)
接收对象中: 100% (4770015/4770015), 1.08 GiB | 2.82 MiB/s, 完成.
处理 delta 中: 100% (3936851/3936851), 完成.
检查连接... 完成。
正在检出文件: 100% (60671/60671), 完成.
mo@mo-N55SL:~/project$ git clone https://TeeFirefly@gitlab.com/TeeFirefly/prebuilts.git
正克隆到 'prebuilts'...
remote: Counting objects: 308, done.
remote: Compressing objects: 100% (120/120), done.
remote: Total 308 (delta 154), reused 308 (delta 154)
接收对象中: 100% (308/308), 26.19 MiB | 4.50 MiB/s, 完成.
处理 delta 中: 100% (154/154), 完成.
检查连接... 完成。
mo@mo-N55SL:~/project$
```

注：内核与编译工具放在同一目录下

4、kernel 编译

linux 内核采用的 config 文件为 firefly_linux_defconfig, dts 文件为 rk3399-firefly-mini -linux.dts,

config 文件在～/linux-kernel/arch/arm64/configs 里面，

dts 文件在～/ linux-kernel/arch/arm64/boot/dts/rockchip 里面

编译内核

cd ~/project/linux-kernel

make ARCH=arm64 firefly_linux_defconfig

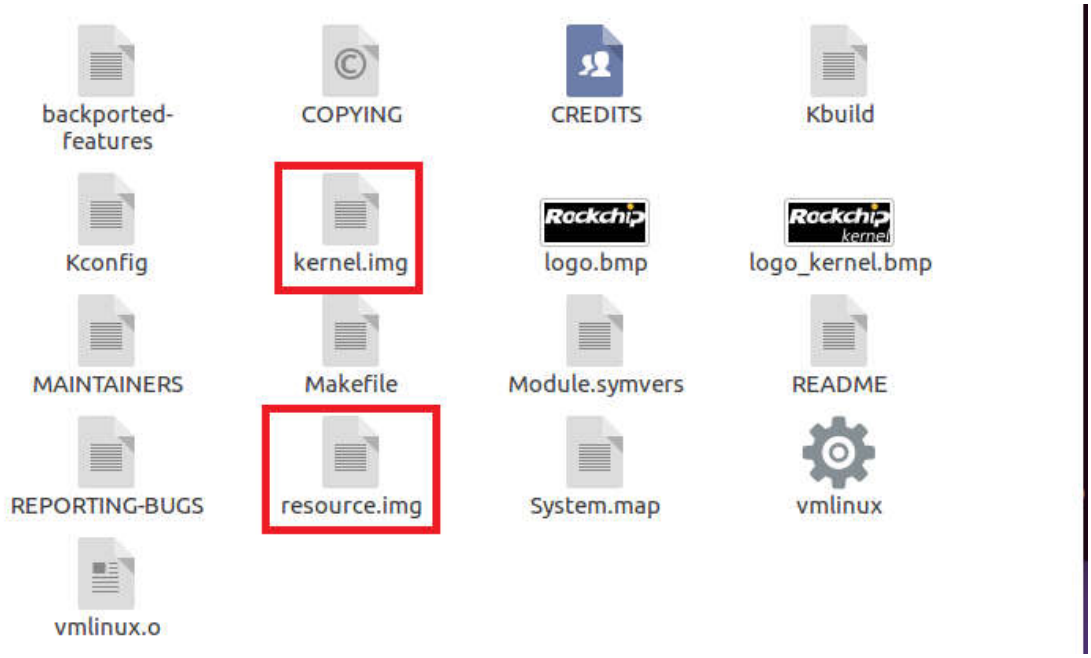
```
mo@mo-N55SL:~$ cd ~/project/linux-kernel
mo@mo-N55SL:~/project/linux-kernel$ make ARCH=arm64 firefly_linux_defconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  SHIPPED scripts/kconfig/zconf.tab.c
  SHIPPED scripts/kconfig/zconf.lex.c
  SHIPPED scripts/kconfig/zconf.hash.c
  HOSTCC  scripts/kconfig/zconf.tab.o
  HOSTLD  scripts/kconfig/conf
#
# configuration written to .config
#
mo@mo-N55SL:~/project/linux-kernel$
```

make ARCH=arm64 rk3399-firefly-linux.img -j8

```
mo@mo-N55SL:~/project/linux-kernel$ make ARCH=arm64 rk3399-firefly-linux.img -j8
scripts/kconfig/conf  --silentoldconfig Kconfig
  HOSTCC  scripts/basic/bin2c
  WRAP    arch/arm64/include/generated/asm/bug.h
  WRAP    arch/arm64/include/generated/asm/checksum.h
  WRAP    arch/arm64/include/generated/asm/bugs.h
  WRAP    arch/arm64/include/generated/asm/clkdev.h
  WRAP    arch/arm64/include/generated/asm/cputime.h
  WRAP    arch/arm64/include/generated/asm/current.h
  CHK     include/config/kernel.release
  WRAP    arch/arm64/include/generated/asm/delay.h
  WRAP    arch/arm64/include/generated/asm/div64.h
  WRAP    arch/arm64/include/generated/asm/dma.h
  WRAP    arch/arm64/include/generated/asm/dma-contiguous.h
  WRAP    arch/arm64/include/generated/asm/early_ioremap.h
  WRAP    arch/arm64/include/generated/asm/emergency-restart.h
  WRAP    arch/arm64/include/generated/asm/ftrace.h
  WRAP    arch/arm64/include/generated/asm/errno.h
  WRAP    arch/arm64/include/generated/asm/hw_irq.h
  WRAP    arch/arm64/include/generated/asm/ioctl.h
  WRAP    arch/arm64/include/generated/asm/ioctls.h
  WRAP    arch/arm64/include/generated/asm/ipcbuf.h
  WRAP    arch/arm64/include/generated/asm/irq_regs.h
  WRAP    arch/arm64/include/generated/asm/kdebug.h
```

```
  LD      drivers/usb/host/xhci-hcd.o
  LD      drivers/usb/host/xhci-plat-hcd.o
  LD      drivers/usb/host/built-in.o
  LD      drivers/usb/built-in.o
  LD      drivers/built-in.o
  LINK    vmlinux
  LD      vmlinux.o
  MODPOST vmlinux.o
  GEN     .version
  CHK     include/generated/compile.h
  UPD     include/generated/compile.h
  CC      init/version.o
  LD      init/built-in.o
  KSYM    .tmp_kallsyms1.o
  KSYM    .tmp_kallsyms2.o
  LD      vmlinux
  SORTEX  vmlinux
  SYSMAP  System.map
  OBJCOPY arch/arm64/boot/Image
  Image:  kernel.img is ready
Pack to resource.img successed!
  Image:  resource.img (with rk3399-firefly-linux.dtb logo.bmp logo_kernel.bmp)
is ready
mo@mo-N55SL:~/project/linux-kernel$
```

编译完成后，会在 linux-kernel 目录下生成 kernel.img 和 resource.img.



5、创建 boot.img

创建内存盘

内核启动时会加载内存盘作为初始的根文件系统，再加载实际的根存储设备，最后切换过去。因为开发板使用的是 eMMC 存储，不需要特别的驱动，因此实际上可以跳过此步。但内存盘可以做得非常灵活和强大，例如可以做多系统启动。

git clone -b for-kernel_4.4 https://github.com/TeeFirefly/initrd.git

make -C initrd

```
mo@mo-N55SL:~/project/linux-kernel$ git clone -b for-kernel_4.4 https://github.com/TeeFirefly/initrd.git
正克隆到 'initrd'...
remote: Counting objects: 313, done.
remote: Total 313 (delta 0), reused 0 (delta 0), pack-reused 313
接收对象中: 100% (313/313), 2.95 MiB | 1.25 MiB/s, 完成.
处理 delta 中: 100% (77/77), 完成.
检查连接... 完成。
mo@mo-N55SL:~/project/linux-kernel$ make -C initrd
make: Entering directory '/home/mo/project/linux-kernel/initrd'
find . ! -path "./.git*" \
       ! -path "./README.md" \
       ! -path "./Makefile" \
| cpio -H newc  -ov | gzip > ../initrd.img
.
./conf
./conf/modules
./conf/arch.conf
./conf/initramfs.conf
./lib
./lib/modules
./lib/modules/rk30xxnand_ko.ko.3.0.36+
./lib/modules/rk30xxnand_ko.ko.3.10.0
./lib/klibc-V6ctNcIsqIUASl2L3t2-3pw4Hvs.so
./lib/libfuse.so.2
./lib/libdevmapper.so.1.02.1
./lib/ld-linux.so.3
./lib/arm-linux-gnueabi
./lib/arm-linux-gnueabi/libc.so.6
./lib/arm-linux-gnueabi/libcom_err.so.2
./lib/arm-linux-gnueabi/libselinux.so.1
./lib/arm-linux-gnueabi/libe2p.so.2
./lib/arm-linux-gnueabi/libntfs-3g.so.831
./lib/arm-linux-gnueabi/libpthread.so.0
./lib/arm-linux-gnueabi/libext2fs.so.2
./lib/arm-linux-gnueabi/libudev.so.0
```

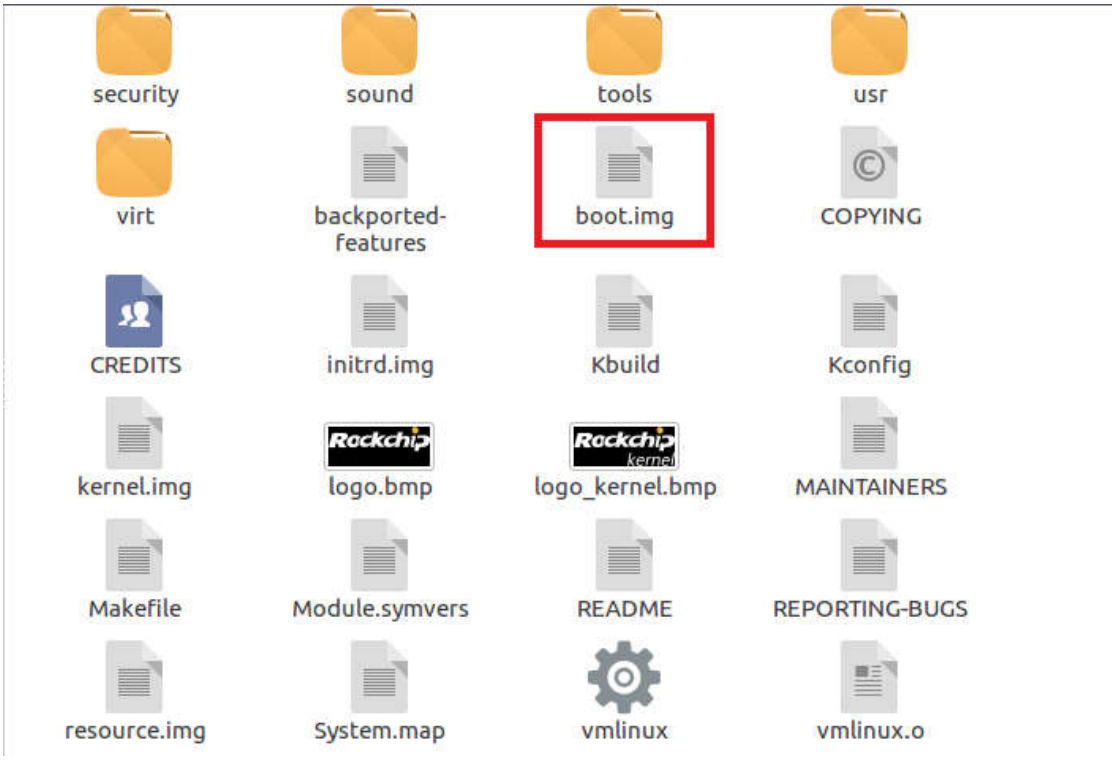
打包内核和内存盘

将 kernel 和 initrd 打包成 boot.img：

mkbootimg --kernel arch/arm64/boot/Image --ramdisk initrd.img --second resource.img -o boot.img

```
./etc/console-setup
./etc/console-setup/cached.kmap.gz
./etc/console-setup/Uni2-Fixed16.psf
./etc/ld.so.conf.d
./etc/ld.so.conf.d/arm-linux-gnueabi_GL.conf
./etc/ld.so.conf.d/arm-linux-gnueabi_EGL.conf
./etc/ld.so.conf.d/arm-linux-gnueabi.conf
./etc/ld.so.conf.d/libc.conf
9091 块
make: Leaving directory '/home/mo/project/linux-kernel/initrd'
mo@mo-N55SL:~/project/linux-kernel$ mkbootimg --kernel arch/arm64/boot/Image --ramdisk initrd.img --second resource.img -o boot.img
mo@mo-N55SL:~/project/linux-kernel$
```

编译完成后，会在 linux-kernel 目录下生成 boot.img.

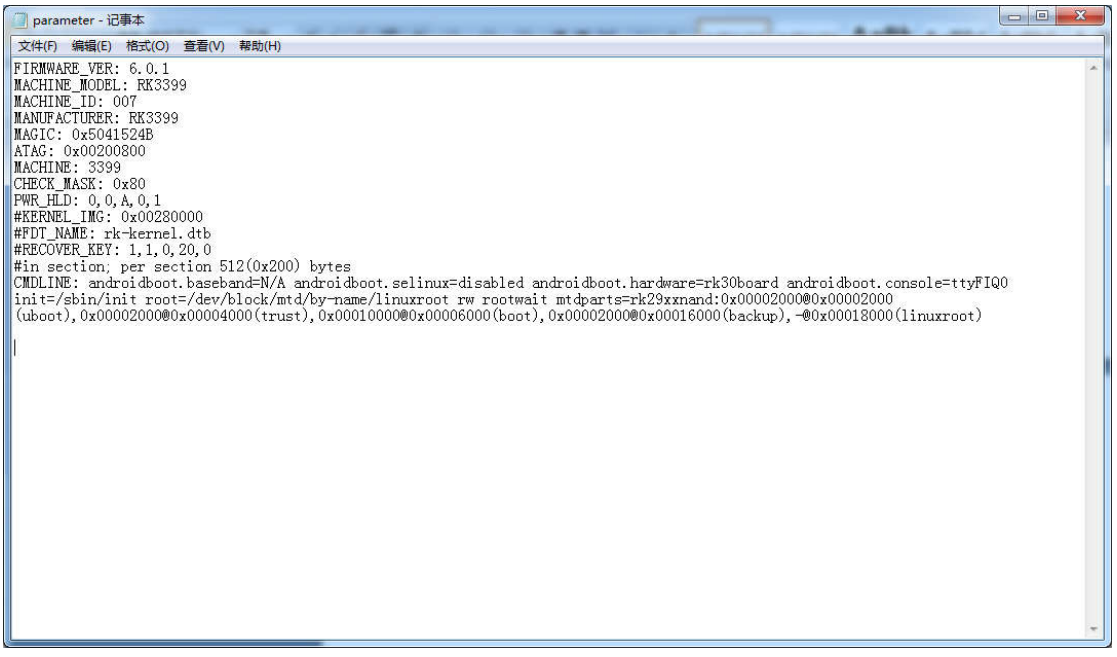


6、修改 parameter 文件

在 Windows 下新建一个 parameter.txt 文件，拷贝以下内容进去，关于 parameter 文件内容信息请参考《Rockchip Parameter File Format Ver1.3》文件：

FIRMWARE_VER: 6.0.1
MACHINE_MODEL: RK3399
MACHINE_ID: 007
MANUFACTURER: RK3399
MAGIC: 0x5041524B
ATAG: 0x00200800
MACHINE: 3399
CHECK_MASK: 0x80
PWR_HLD: 0,0,A,0,1
#KERNEL_IMG: 0x00280000
#FDT_NAME: rk-kernel.dtb
#RECOVER_KEY: 1,1,0,20,0
#in section; per section 512(0x200) bytes
CMDLINE: androidboot.baseband=N/A androidboot.selinux=disabled androidboot.hardware=rk30board androidboot.console=ttyFIQ0 init=/sbin/init root=/dev/block/mtd/by-name/linuxroot rw rootwait mtdparts=rk29xxnand:0x00002000@0x00002000(uboot),0x00002000@0x00004000(trust),0x00010000@0x00006000(boot),0x00002000@0x00016000(backup),-@0x00018000(linuxroot)

如图示：



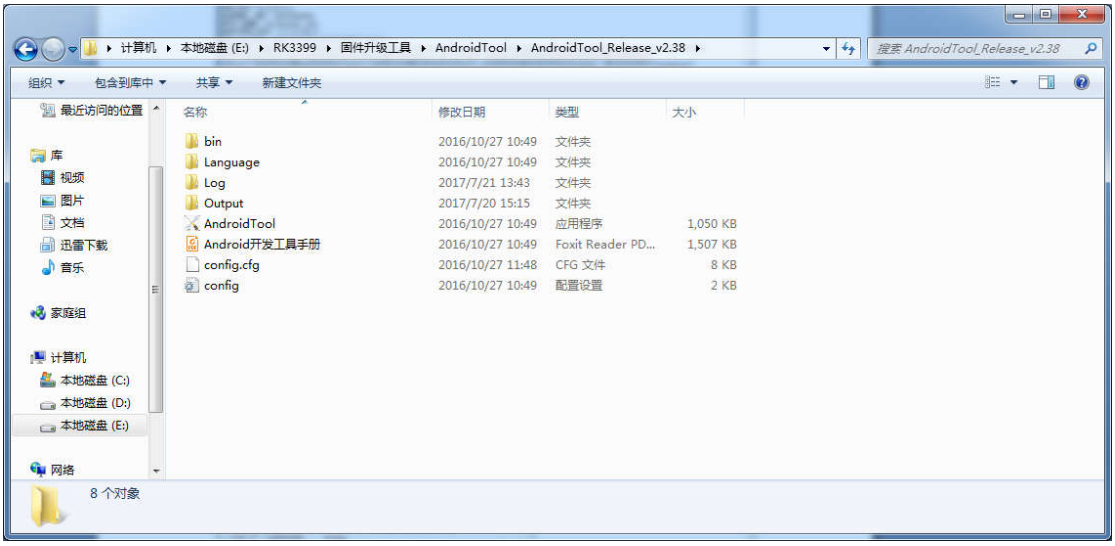
/**********

这里说明一下，root=/dev/block/mtd/by-name/linuxroot 后面加 rw rootwait，两个参数，rw 是声明启动权限，即以读写方式启动；rootwait 是指等待设备 /dev/block/mtd/by-name/linuxroot 设备就绪后才尝试挂载 rootfs。如果没有此参数，linux 内核启动时可能会在存储设备尚未就绪是就尝试挂载 rootfs，此时肯定

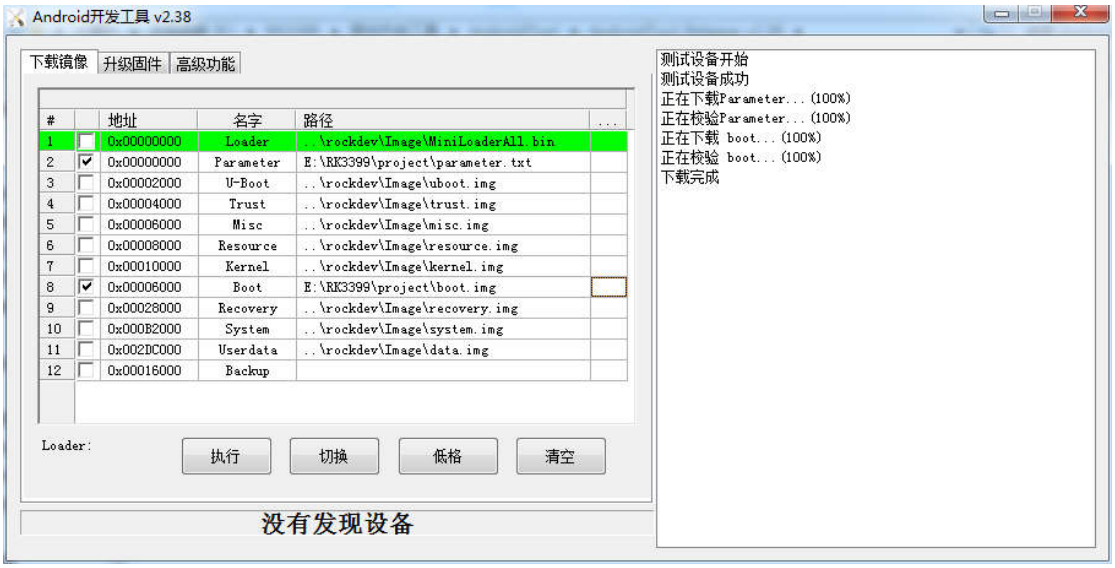
挂载失败，那么启动也就失败了。
**********/

7、烧写到设备

运行 AndroidTool_Release_v2.38 目录里面的 AndroidTool.exe



选择生成的 boot.img 和修改过的 parameter 文件，分别烧写到 "boot" 和 "parameter" 分区，点击“执行”，则可完成内核的更新。



顺利进入界面，内核编译完成。